Skeleton Clustering: Graph-Based Approach for Dimension-Free Density-Aided Clustering

Zeyu Wei
Department of Statistics
University of Washington
Seattle, WA 98195
zwei5@uw.edu

Yen-Chi Chen
Department of Statistics
University of Washington
Seattle, WA 98195
yenchic@uw.edu

Abstract

Density-based clustering can identify clusters with irregular shapes and has intuitive interpretations, but struggles with large-dimensional data due to the curse of dimensionality. We introduce a graph-based clustering framework called *Skeleton Clustering* to adopt density-based clustering idea to multivariate and even high-dimensional data. The proposed framework constructs a graph representation of the data as a first step and combines prototype methods, density-based clustering, and hierarchical clustering. We propose surrogate density measures based on the skeleton graph that are less dependent on the dimension and have meaningful geometric interpretations. We show by empirical studies that the proposed skeleton clustering method leads to reliable clusters in multivariate and even high-dimensional data with irregular shapes.

1 Introduction

Density-based clustering [Azzalini and Torelli, 2007, Menardi and Azzalini, 2014, Chacón, 2015] is a popular framework to group observations into clusters defined based on the underlying probability density function (PDF). In practice, when the PDF is unknown, it is estimated via the random sample and the estimated PDF is used to obtain the resulting clusters. Many clustering methods have been proposed within the framework of density-based clustering. The mode clustering [Li et al., 2007, Chacón and Duong, 2013, Chen et al., 2016] finds clusters via the local modes of the underlying PDF. When the kernel density estimator (KDE) is used for density estimation, the mode clustering can be done easily via the mean-shift algorithm [Fukunaga and Hostetler, 1975, Cheng, 1995, Carreira-Perpinán, 2015]. Another famous density-based clustering approach is the level-set clustering [Cuevas et al., 2000, 2001, Mason et al., 2009, Rinaldo et al., 2012], which creates clusters as the connected components of high density regions. The well-known DBSCAN method [Ester et al., 1996] is also a special case of level set clustering. Moreover, the cluster tree [Stuetzle and Nugent, 2010, Chaudhuri and Dasgupta, 2010, Chaudhuri et al., 2014, Eldridge et al., 2015, Kim et al., 2016] is a density-based clustering approach combining information from both modes and level-sets. This method creates a tree structure with each leaf represents a mode and the tree describes the evolution of level-set clusters at different density levels.

Compared to the classical k-means clustering [Lloyd, 1982, Hartigan and Wong, 1979, Pollard, 1982] and the model-based clustering methods [Fraley and Raftery, 2002], a density-based clustering approach is capable of finding clusters with irregular shapes and gives an intuitive interpretation based on the underlying PDF. Furthermore, defining clusters based on the density function makes it possible to view the clustering problem as an estimation problem: the clusters from the true PDF are the parameters of interest and the estimated clusters are sample quantities utilized for approximation.

NeurIPS 2022 New Frontiers in Graph Learning Workshop (NeurIPS GLFrontiers 2022).

Although density-based clustering enjoys many advantages, it has a fundamental limitation that, due to the curse of dimensionality, density estimation does not scale well with the dimension. Specifically, the convergence rate of a density estimator is $O_P(n^{-\frac{2}{4+d}})$ under usual smoothness conditions [Scott, 2015, Wasserman, 2006], which is slow when d is large. In this work, we learn a graph representation of the data to overcome the curse of dimensionality when applying density-based clustering approach. In particular, we merge protoclusters [Peterson et al., 2018, Fred and Jain, 2005, Maitra, 2009, Baudry et al., 2010, Shin et al., 2019, Hennig, 2010] through graph-based density-aided criterion. Our idea can be summarized as follows. We first find a large set of protoclusters (called *knots*) by running k-means clustering. Nearby knots are then connected by edges to form a graph that we call the *skeleton*. The similarities between connected knots are measured based on the skeleton through some density-aided criteria that are estimable even in high dimensions. Finally, we merge knots according to a linkage criterion to create the final clusters. Because the construction involves creating a *skeleton* representation of the data, we call this method *Skeleton Clustering*.

To illustrate the limitation of the classical approaches and to highlight the effectiveness of skeleton clustering, we conduct a simple simulation in Figure 1. It is a $d = 200$ dimensional data consisted of five components with non-spherical shapes. The actual structure is in 2-dimensional space as illustrated in Figure 1. We add Gaussian noises in other dimensions to make it a $d = 200$ dimensional data (see Section 5 for more details). Traditional k-means and spectral clustering fail to find the five components and mean shift algorithm cannot form clusters due to the high dimensionality of the data. However, our proposed method (bottom-right panel) can successfully recover the underlying five components.

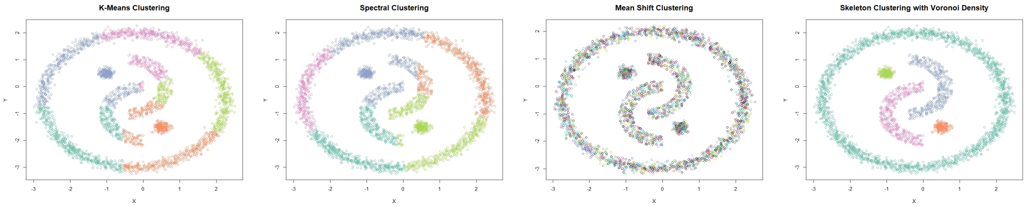


Figure 1: Yinyang Data with dimension 200. On the bottom-right is the clustering result of the skeleton clustering with the proposed Voronoi density similarity measure.

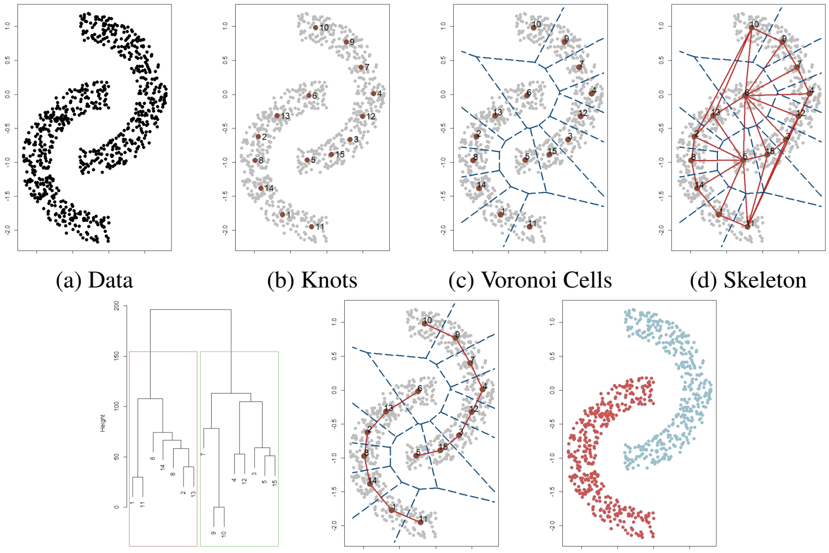
2 Skeleton Clustering Framework

In this section we formally introduce the skeleton clustering framework. Let $\mathbb{X} = \{X_1, \ldots, X_n\}$ be a random sample from an unknown distribution with density p supported on a compact set $\mathcal{X} \in \mathbb{R}^d$. The goal of clustering is to partition $\mathbb{X}$ into clusters $\mathbb{X}_1, \ldots \mathbb{X}_S$, where S is the final number of clusters.

Algorithm 1 Skeleton clustering

```
**Input:** Observations $X_1, \cdots, X_n$, final number of clusters $S$.
1. **Knot construction.** Perform $k$-means clustering with a large number of $k$; the centers are the knots (Section 2.1).
2. **Edge construction.** Apply approximate Delaunay triangulation to the knots (Section 2.2).
3. **Edge weights construction.** Add weights to each edge using either Voronoi density, Face density, or Tube density similarity measure (Section 3).
4. **Knots segmentation.** Use linkage criterion to segment knots into $S$ groups based on the edge weights (Section 2.4).
5. **Assignment of labels.** Assign a cluster label to each observation based on which knot-group the nearest knot belongs (Section 2.5).
```

A summary of the skeleton clustering framework is provided in Algorithm 1. Figure 2 illustrates the overall procedure of the skeleton clustering method. Starting with a collection of observations (panel (a)), we first find knots, the representative points of the entire data (panel (b)). Then we compute the corresponding Voronoi cells induced by the knots (panel (c)) and the edges associating the nearby



(a) Data (b) Knots (c) Voronoi Cells (d) Skeleton

(e) Dendrogram (f) Segmentation (g) Clustering

Figure 2: Skeleton Clustering illustrated by Two Moon Data (d=2).

Voronoi cells (panel (d)). For each edge in the graph, we compute a density-aided similarity measure that quantifies the closeness of each pair of knots. For the next step we segment knots into groups based on a linkage criterion (single linkage in this example), leading to the dendrogram in panel (e). Finally, we choose a threshold that cuts the dendrogram into $S = 2$ clusters (panel (f)) and assign cluster label to each observation according to the knot-cluster that it belongs to (panel (g)).

In summary, the skeleton clustering consists of the following five steps: (1) Knots construction, (2) Edges construction, (3) Edge weights construction, (4) Knots segmentation, and (5) Assignment of labels. In what follows in this section, we provide a detailed description of each step except Step 3. Step 3 is the key step in our clustering framework where we incorporate the information from the underlying density for clustering in a less dimension-dependent way and we defer the detailed discussion of Step 3 to Section 3. We include a short analysis on the computational complexity of our skeleton clustering framework in Appendix B.

2.1 Knots Construction

The knots are constructed as representative points in the data that can help measure similarities between regions in the later stage. The knots can be viewed as landmarks inside the data where we can shift our focus from the entire data to these local locations. A simple but reliable approach for constructing knots is the k-means algorithm. We apply the k-means algorithm with a large number $k \gg S$ the desired number of final clusters, and this procedure behaves like overfitting the k-means. Notably, we do not use k-means procedure to obtain final clustering, but instead we use it as a intermediate step to find concise representations of the original data.

The number of knots k is a key parameter in the knots construction step. It controls the trade-off between the quality of the data representation and the reliability of each knot. More knots can give better representation of the data, but, if we have too many knots, the number of observation per knot will be small, so the uncertainty in estimation in the later stage will be large. We find that a simple reference rule for k to be around $\sqrt{n}$ works well in our empirical studies (Section D.1). In practice, it is also advisable to prune knots with a small number of corresponding observations because the density-aided weights (in Step 3, Section 3) are estimated locally by the data belonging to each pair of knots. Knots with a few data points can lead to unstable similarity measurements and unreliable final clustering. Moreover, to take care of observations in the low-density areas that could cause problems for the k-means clustering, one may first pre-process or denoise the data by removing observations in the low-density area and then apply the k-means clustering to find out the knots.

In this work we use overfitting k-means as the default way for knots construction, but there are alternative approaches to find knots such as subsampling, the coreset construction methods [Bachem et al., 2017, Turner et al., 2020], and the Self-Organizing Maps (SOM) [Heskes, 2001]. We show in Appendix D.2 that the SOM can also be used to find knots but requires more careful treatments such as removing knots with few or even no observations and the performance is slightly worse than that of the overfitting k-means. The k-medians algorithm can be another alternative method but it gave an unstable result when the dimension is large. Therefore, we choose to use the overfitting k-means algorithm in this work and recommend using it in practice.

2.2 Edges Construction

With the constructed knots, our next step is to find the edges connecting them. Let $c_1, \cdots, c_k$ be the given knots and we use $\mathcal{C} = \{c_1, \cdots, c_k\}$ to denote the collection of them. We add an edge between a pair of knots if they are neighbors, with the neighboring condition being that the corresponding Voronoi cells [Voronoi, 1908] share a common boundary. The Voronoi cell, or Voronoi region, $\mathbb{C}_j$, associated with a knot c_j is the set of all points in $\mathcal{X}$ whose distance to c_j is the smallest compared to other knots (See Figure 3). That is,

$$\mathbb{C}_j = \{x \in \mathcal{X} : d(x, c_j) \leq d(x, c_\ell) \ \forall \ell \neq j\}, \tag{1}$$

where $d(x, y)$ is the usual Euclidean distance. Therefore, we add an edge between knots (c_i, c_j) if $\mathbb{C}_i \cap \mathbb{C}_j \neq \emptyset$. Such resulting graph is the Delaunay triangulation [Delaunay, 1934] of the set of knots $\mathcal{C}$ and we denote it as $DT(\mathcal{C})$. In a nutshell, the skeleton graph in our framework is given by the Delaunay triangulation of $\mathcal{C}$.

The Delaunay triangulation graph is conceptually intuitive and is utilized by some clustering methods to identify connected components [Azzalini and Torelli, 2007, Scrucca, 2016], but empirically the computational complexity of the exact Delaunay triangulation algorithm has an exponential dependence on the ambient dimension d [Amenta et al., 2007, Chazelle, 1993]. Given our multivariate and even high-dimensional data setting, exact Delaunay triangulation is empirically unfavorable. Therefore, in practice, we approximate the exact Delaunay Triangulation with $\hat{DT}(\mathcal{C})$ by examining the 2-nearest knots of the sample data points. The key observation is that, if the Voronoi cells of two knots c_i, c_j share a boundary, there is a non-empty region of points whose 2-nearest knots are c_i, c_j. Consequently, for approximation, we query the two nearest knots for each data point and have an edge between c_i, c_j if there is at least one data point whose two nearest neighbors are c_i, c_j. The complexity of the neighbor search depends linearly on the dimension d, which is desirable for high-dimensional setting [Weber et al., 1998], and this sample-based approximation to the Delaunay Triangulation has reliable empirical performance.

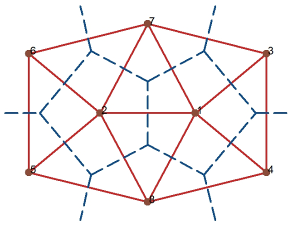


Figure 3: Voronoi Tessellation as blue dashed lines and Delaunay Triangulation by red solid lines.

2.3 Edge Weight Construction

Given the constructed edges and knots, we assign each edge a weight that represents the similarity between the pair of knots. In this work, we propose some novel density-aided quantities as the edge weights. Since the description of the similarity measures is more involved, we defer the detailed discussion of the similarity measures to Section 3. It is worth noting here that the similarity measures proposed in this work are estimated based on surrogates of the underlying density function (hence density-aided) and the estimation procedure has minimal dependence on the ambient dimension. Therefore, the estimations of the newly proposed similarity measures are reliable even under high-dimensional settings.

2.4 Knots Segmentation

Given the weighted skeleton graph, the next step is to partition the knots into the desired number of final clusters, and we apply hierarchical clustering with the inverses of the similarity measures as the distance. The choice of linkage criterion for hierarchical clustering may depend on the underlying geometric structure of the data. Empirically, single linkage gives reliable clustering results when the components are well-separated, but average linkage works better when there are overlapping clusters of approximately spherical shapes. Therefore, in practice, such choice of linkage should be made base on some exploratory understanding of the data structure, and experimenting with different linkage methods is computationally tractable as only the knots need to be segmented.

The number of final clusters S is an essential parameter for the hierarchical clustering procedure but can be unknown. The dendrograms given by hierarchical clustering can be a helpful tool in this situation, displaying the clustering structure at different resolutions. Consequently, analysts can experiment with different numbers of final clusters and choose a cut that preserves the meaningful structures based on the dendrograms, which takes little extra computation. However, it is worth pointing out that with the presence of noisy data points, setting the final number S to be larger than the true number of meaningful components may be needed to achieve better clustering results.

2.5 Assignment of Labels

In the previous step, we have created S groups of knots and each group has a cluster label. To pass the cluster membership to each observation, we assign a hard clustering label to each observation according to which group its nearest knot belongs. For instance, if an observation X_i is closest to knot c_j and c_j belongs to cluster ℓ, we assign cluster membership label ℓ to observation X_i.

3 Density-Aided Edge Weights

To incorporate the information of density into clustering, we calculate the edge weights based on the underlying density function. However, the conventional notion of PDF is not feasible in multivariate or even high-dimensional data due to the curse of dimensionality. To resolve this issue, we introduce three density-related quantities that are estimable even when the dimension is high.

3.1 Voronoi Density

The *Voronoi density (VD)* measures the similarity between a pair of knots (c_j, c_ℓ) based on the number of observations whose 2-nearest knots are c_j and c_ℓ. We start with defining the Voronoi density based on the underlying probability measure and then introduce its sample analog. Given a metric d on $\mathbb{R}^d$, the 2-Nearest-Neighbor (2-NN) region of a pair of knots (c_j, c_ℓ) is defined as

$$A_{j\ell} = \{x \in \mathcal{X} : d(x, c_i) > \max\{d(x, c_j), d(x, c_\ell)\}, \forall i \neq j, \ell\}. \tag{2}$$

In this work we take $d(.,.)$ to be usual Euclidean distance and use $||.||$ to denote the Euclidean norm. An example 2-NN region of a pair of knots is illustrated in Figure 4.

Following the idea of density-based clustering, two knots c_j, c_ℓ belongs to the same clusters if they are in a connected high-density region, and we would expect the 2-NN region of c_j, c_ℓ to have a high probability measure. Hence, the probability $\mathbb{P}(A_{j\ell}) = P(X_1 \in A_{j\ell})$ can measure the association between c_j and c_ℓ. Based on this insight, the Voronoi density measures the edge weight of (c_j, c_ℓ) with

$$S_{j\ell}^{VD} = \frac{\mathbb{P}(A_{j\ell})}{\|c_j - c_\ell\|}. \tag{3}$$

Namely, we divide the probability of in-between region by the mutual Euclidean distance. The division of the distance adjusts for the fact that 2-NN regions have different sizes and provides more weights to edges between knots close in distance. However, such division makes the Voronoi density to be in the unit of $^1/_{\|c_j - c_\ell\|}$ and hence can be scale-dependent.

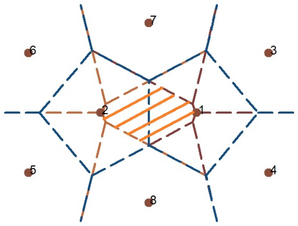


Figure 4: Orange shaded area illustrates the 2-NN region of knots $1, 2$.

In practice we estimate $S_{j\ell}^{VD}$ by a sample average. Specifically, the numerator $\mathbb{P}(A_{j\ell})$ is estimated by $\hat{P}_n(A_{j\ell}) = \frac{1}{n}\sum_{i=1}^{n} I(X_i \in A_{j\ell})$ and the final estimator for the VD is

$$\hat{S}_{j\ell}^{VD} = \frac{\hat{P}_n(A_{j\ell})}{\|c_j - c_\ell\|}. \tag{4}$$

Note that here we are assuming that $c_1, \cdots, c_k$ as given beforehand. In sample version, we replace them by the sample analog $\hat{c}_1, \cdots, \hat{c}_k$ and replace the region $A_{j\ell}$ by $\hat{A}_{j\ell}$.

The Voronoi density can be computed in a fast way. The numerator, which only depends on 2-nearest-neighbors calculation, can be computed efficiently by the k-d tree algorithm [Bentley, 1975]. For high-dimensional space, space partitioning search approaches like the k-d tree can be inefficient but a direct linear search still gives a short run-time [Weber et al., 1998], and with a large number of observations approximate nearest neighbor algorithms can be incorporated. The denominator requires distance calculation and can be burdensome in high-dimensional settings, but note that we only need to calculate the distance for edges present in $\hat{DT}(\mathcal{C})$, which is far less than $k(k-1)/2$, where k is the number of knots. Hence, the calculation of VD can be carried out in a fast way even for high-dimensional data with a large sample size.

3.2 Face Density

Here we present another density-based quantity to measure the similarity between two knots. Since the Voronoi cell of a knot describes the associated region, a natural way to measure the similarity between two knots is to investigate the shared boundary of the corresponding Voronoi cells. If two knots are highly similar, we would expect the boundary to lie in a high-density region and to be surrounded by many observations. Based on this idea, we define the *Face Density (FD)* as the integrated PDF over the "face" (boundary) region. Note that, although the density is involved in FD, by integrating over the face region the problem reduces to a 1-dimensional density estimation task regardless of the dimension of the ambient space. Formally, let the face region between two knots c_j, c_ℓ be $F_{j\ell} = \mathbb{C}_j \cap \mathbb{C}_\ell$. At the population level, the FD is defined as

$$S_{j\ell}^{FD} = \int_{F_{j\ell}} p(x)\mu_{d-1}(dx) = \int_{F_{j\ell}} d\mathbb{P}(x), \tag{5}$$

where $\mu_m(dx)$ denotes the m-dimensional volume measure.

To estimate the FD, we utilize the idea of kernel smoothing in combination with data projection. By the construction of the Voronoi diagram, the boundary of two Voronoi cells is orthogonal to the line passing through the two corresponding knots (called the 'central line') and intersects the central line at the middle point regardless of the dimension of the data (see Figure 3 for reference). Therefore, we estimate the FD by first projecting the observations onto the central line and then using the 1-dimensional kernel density estimator(KDE) to evaluate the density at the midpoint. Specifically, fix two knots c_j, c_ℓ, let $\mathbb{C}_j, \mathbb{C}_\ell$ be the corresponding Voronoi cells, and denote $\Pi_{j\ell}(x)$ as the projection of $x \in \mathcal{X}$ onto the central line passing through c_j and c_ℓ, we define the estimator $\hat{S}_{j\ell}^{FD}$ to be

$$\hat{S}_{j\ell}^{FD} = \frac{1}{nh} \sum_{X_i \in \mathbb{C}_j \cup \mathbb{C}_\ell} K\left(\frac{\Pi_{j\ell}(X_i) - (c_\ell + c_j)/2}{h}\right) \tag{6}$$

where K is a smooth, symmetric kernel function (e.g. Gaussian kernel) and $h > 0$ is the bandwidth that controls the amount of smoothing. It is noteworthy that, while the conventional kernel smoothing suffers from the curse of dimensionality [Chen, 2017, Chacón et al., 2011, Wasserman, 2006], the kernel estimator in equation (6) bypasses it.

While FD is conceptually appealing, the characterization of the face between two Voronoi cells could be challenging since the shapes of the boundaries can be irregular. We propose a measure similar to the Face density measure but has a predefined regular shape and discuss it in detail in Appendix A.

4 Asymptotic Theory of Edge Weight Estimation

In this section, we focus on the theoretical properties of the similarity measures to theoretically explain the effectiveness of the newly proposed density-aided similarity measures. We assume the set of knots $\mathcal{C} = \{c_1, \ldots, c_k\}$ is given and non-random to simplify the analysis because (1) it is hard to quantify k-means uncertainty, and (2) with large k, it is extremely likely for k-means to stuck within the local minimum. Note that this implies the corresponding Voronoi cells $\mathbb{C} = \{\mathbb{C}_1, \ldots, \mathbb{C}_k\}$ and the 2-NN regions $\{A_{j\ell}\}_{j,\ell=1,\ldots,k,j\neq\ell}$ (Equation 2) of all pairs of knots are fixed as well. We allow $k = k_n$ to grow with respect to the sample size n. Theoretical results for Voronoi density are described in this section with proofs included in Appendix C.

4.1 Voronoi Density Consistency

We start with the convergence rate of the VD. We consider the following condition:

- **(B1)** There exists a constant c_0 such that the minimal knot size $\min_{(j,\ell)\in E} \mathbb{P}(A_{j\ell}) \geq \frac{c_0}{k}$ and $\min_{(j,\ell)\in E} \|c_j - c_\ell\| \geq \frac{c_0}{k^{1/d}}$.

where $(j,\ell) \in E$ means that there is an edge between knots c_j, c_ℓ in the Delaunay Triangulation. Condition (B1) is a condition requiring that no Voronoi cell $A_{j\ell}$ has a particularly small size and all edges have sufficient length. This condition is mild because when the dimension of data d is fixed, the total number of edges in the Delaunay triangulation of k points scale at rate $O(k)$. Because the volume shrinks at rate $O(k^{-1})$, the distance is expected to shrink at rate $O(k^{-1/d})$.

Theorem 1 (Voronoi Density Convergence). Assume (B1). Then for any pair $j \neq \ell$ that shares an edge, the similarity measure based on the Voronoi density satisfies

$$\left| \frac{\hat{S}^{VD}_{j\ell}}{S^{VD}_{j\ell}} - 1 \right| = O_p\left(\sqrt{\frac{k}{n}}\right), \tag{7}$$

$$\max_{j,\ell} \left| \frac{\hat{S}^{VD}_{j\ell}}{S^{VD}_{j\ell}} - 1 \right| = O_p\left(\sqrt{\frac{k}{n}} \log k\right), \tag{8}$$

when $n \to \infty, k \to \infty, \frac{n}{k} \to \infty$.

Theorem 1 provides the convergence rates of the sample-based Voronoi density to the population version Voronoi density. This result is reasonable because when the knots $\mathcal{C}$ are given, the randomness in the sample-based Voronoi density is just the empirical proportion in each cell, so it is a square-root-rate estimator based on the effective local sample size n/k. Consequentially, Theorem 1 suggests that estimating the Voronoi density is easy in multivariate case when the knots are given–there is no dependency with respect to the ambient dimension. The extra $\log k$ factor in the uniform bound (Equation 8) comes from the Gaussian concentration bounds.

4.2 Performance Guarantee for Voronoi Density

We provide below a performance guarantee in terms of the adjusted Rand Index [Rand, 1971, Hubert and Arabie, 1985] for skeleton clustering with Voronoi density edge similarity. To simplify the problem, we define the true clusters as the connected components of the skeleton graph with edges having true Voronoi density similarities $S^{VD}_{j\ell}$ over a known threshold $\tau > 0$. We show below that cutting the skeleton graph based on estimated edge similarities at the same threshold τ recovers the true clustering with a high probability. Since the knots are fixed, the clustering error comes from partitioning knots into the wrong groups, so we will focus on the adjusted Rand Index of clustering the knots. Let the true partition of the knots be $\mathcal{L}^* = \{\mathcal{L}^*_\ell\}_{\ell=1,\dots,L}$, where $\mathcal{L}^*_\ell$ contains all the knot indices belonging to the partition ℓ. Let the partition based on estimated edge similarities be $\hat{\mathcal{L}}$. We assume that

- **(P1)** The true partition $\mathcal{L}^*$ under the threshold τ remains the same when the thresholding level is within $(\tau(1-\varepsilon), \tau(1+\varepsilon))$ for some $\varepsilon > 0$.

This is a mild assumption because when we vary the threshold level τ, only a finite number of value will create a change in the partition. So (P1) holds under almost all values of τ except for a set of Lebesgue measure 0. Let $ARI(\mathcal{L}^*, \hat{\mathcal{L}})$ denotes the adjusted Rand Index of the estimated partition.

Theorem 2 (Adjusted Rand Index Guarantee). Assume (B1) and (P1) and let $p_{min} = \min_{j,\ell} \mathbb{P}(A_{j\ell})$, then

$$\mathbb{P}\left\{ ARI(\mathcal{L}^*, \hat{\mathcal{L}}) < 1 \right\} \leq k(k-1) \exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1-p_{min}) + \frac{1}{3}\varepsilon} \right) \tag{9}$$

Theorem 2 shows that we have a good chance of recovering the "true" clusters defined by the actual Voronoi density. The above bound is derived from the uniform concentration bound of the Voronoi density.

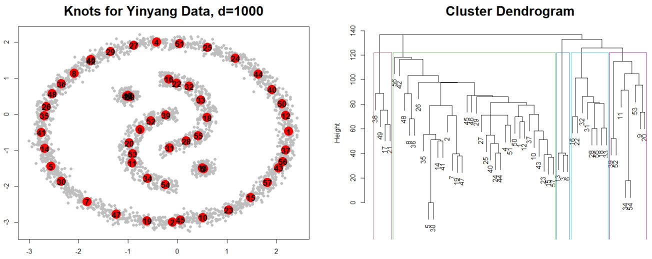


Figure 5: Knots chosen by k-means on Yinyang data and the Dendrogram for single linkage hierarchical clustering with similarity measured by Voronoi density.

5 Simulations

To study the effectiveness of skeleton clustering as a clustering method, we conduct several Monte Carlo experiments. In this section, we present some empirical results to illustrate the performance of skeleton clustering in multivariate and high-dimensional settings (with additional data examples in Appendix E). Generally, our framework with the Voronoi density similarity measure is superior among all the compared clustering methods. We include some additional simulations to support some choices within our framework in Appendix D.

Experimental Setup For the simulations in this section and in Appendix E, when using the skeleton clustering methods, the number of knots is set to be $k = [\sqrt{n}]$ and the knots are chosen by k-means with 1000 random initialization. We select smoothing bandwidth by the normal scale bandwidth selector for the FD and TD, and the radius of TD is set to be the same for all edges with the value chosen as described in Section A. We use single linkage hierarchical clustering when merging knots into final clusters with the true number of final clusters S being provided.

To highlight the importance of density-aided similarity measures, we include a similarity measure called the average distance (AD) for comparison. AD measures the similarity between c_j and c_ℓ using the inverse of the average Euclidean distances between all pairs of observations in the two corresponding Voronoi cells. All simulations are repeated 100 times to obtain the distribution of the empirical performances.

5.1 Yinyang Data Results

The Yinyang dataset is an intrinsically 2-dimensional data containing 5 components: a big outer circle with 2000 uniformly distributed data points, two inner semi-circles each with 200 data points generated as 2D Gaussian with standard deviation 0.1, and two clumps each with 200 data points (generated with the `shapes.two.moon` function with default parameters in the `clusterSim` library in R [Walesiak and Dudek, 2020]). The total sample size is $n = 3200$ and according to our reference rule we choose $k = [\sqrt{3200}] = 57$ knots for the skeleton clustering procedure. To make the data high-dimensional, we include additional variables from a Gaussian distribution with mean 0 and standard deviation 0.1, and we increase the dimension of noise variables so that the total dimensions are $d = 10, 100, 500, 1000$. We present results with larger standard deviations for the noisy variable in Appendix D.7. We empirically compare the following clustering approaches: direct single-linkage hierarchical clustering (SL), direct k-means clustering (KM), spectral clustering (SC), skeleton clustering with average distance density (AD), skeleton clustering with Voronoi density (Voron), skeleton clustering with Face density (Face), and skeleton clustering with Tube density (Tube). Since this is a simulated data, we know that there are exactly 5 clusters and we know which cluster an observation belongs to. The true number of clusters is provided to all the clustering algorithms. We use the adjusted Rand Index to measure the performance of each clustering method.

The results are given in Figure 6. We observe that when dimension increases, traditional methods (SL, KM, SC) fail to give good clustering results while skeleton clustering can generate nearly perfect clustering. Across all the data dimensions, the Voronoi density, the simplest measure among the three proposed similarity measures, gives the best performance in skeleton clustering framework. Average distance density becomes problematic in high-dimensional settings but still gives better

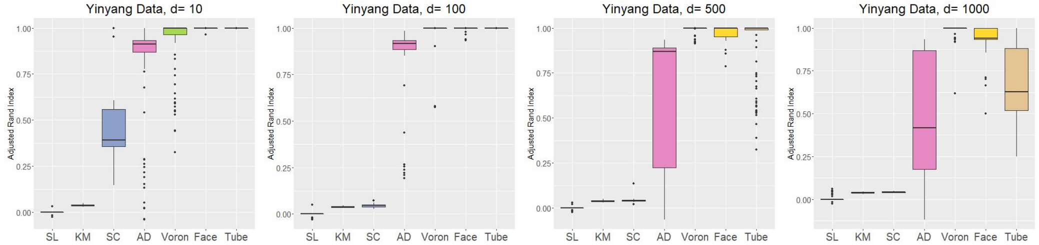


Figure 6: Comparison of the final clustering performance in terms of adjusted Rand Index with different clustering methods on Yinyang Data with dimension 10, 100, 500, and 1000.

performance compared to the classical methods. The fact that all skeleton clustering methods perform better than the traditional methods highlights the effectiveness of using the skeleton clustering framework. Moreover, all three density-aided similarity measures outperform the average distance, which illustrates the power of using density-aided weights in clustering.

6 Conclusion and Discussion

In this work, we introduce the skeleton clustering framework that can handle multivariate and even high-dimensional clustering problems with complex, manifold-based cluster shapes. Our method adopts the density-based clustering idea to the high dimensional regime. The key to bypass the curse of dimensionality is the use of density surrogates such as Voronoi density, Face density, and Tube density that are less sensitive to the dimension. We use empirical analysis to illustrate the effectiveness of the skeleton clustering procedure.

Despite the established results, future works can improve the proposed framework. First, we have some preliminary theoretical results justifying the proposed framework but we need to better account for the randomness of the knots. The randomness of knots can affect the clustering performance because the location of knots directly impact the Voronoi cells, which changes the value of the similarity measures and consequently the cluster label assignments. In particular, observations on the boundary of clusters will be more sensitive to any perturbations on the location of knots. Currently, there are two technical challenges when dealing with random knots. First, the randomness of knots may be correlated with the randomness of estimated edge weight, so the calculation of rates is much more complicated. Second, while there are established theories for k-means algorithm [Graf and Luschgy, 2000, 2002, Hartigan and Wong, 1979], these results only apply to the global minimum of the objective function. In reality, we are unlikely to obtain the global minimum, but instead, our inference is based on a local minimum. It is unclear how to properly derive a theoretical statement based on local minima, so we leave this as future work.

For future directions, the proposed skeleton clustering framework has the potential on anomaly and noise detection and on boundary detection. Other works in graph learning literature can also be incorporated into the proposed framework. Overall, skeleton clustering framework is flexible, and potentially using some new methods at different steps can provide new insights for different data sets even when the data are high-dimensional and large in scale.

References

Nina Amenta, Dominique Attali, and Olivier Devillers. Complexity of delaunay triangulation for points on lower-dimensional polyhedra. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '07, pages 1106–1113, USA, 2007. Society for Industrial and Applied Mathematics. ISBN 9780898716245.

Adelchi Azzalini and Nicola Torelli. Clustering via nonparametric density estimation. *Statistics and Computing*, 17(1):71–80, 2007.

Olivier Bachem, Mario Lucic, and Andreas Krause. Practical coreset constructions for machine learning, 2017.

Jean-Patrick Baudry, Adrian E Raftery, Gilles Celeux, Kenneth Lo, and Raphaël Gottardo. Combining mixture components for clustering. *Journal of Computational and Graphical Statistics*, 19(2):332–353, 2010. doi: 10.1198/jcgs.2010.08111.

Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9): 509–517, September 1975.

Adrian W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

Ryan Remy Brinkman, Maura Gasparetto, Shang Jung Jessica Lee, Albert J. Ribickas, Janelle Perkins, William Janssen, Renee Smiley, and Clay Smith. High-Content Flow Cytometry and Temporal Data Analysis for Defining a Cellular Signature of Graft-Versus-Host Disease. *Biology of Blood and Marrow Transplantation*, 13(6):691–700, jun 2007.

Miguel A Carreira-Perpinán. A review of mean-shift algorithms for clustering. *arXiv preprint arXiv:1503.00687*, 2015.

José E Chacón. A Population Background for Nonparametric Density-Based Clustering. *Statistical Science*, 30 (4):518–532, 2015. doi: 10.1214/15-STS526.

José E Chacón and Tarn Duong. Data-driven density derivative estimation, with applications to nonparametric clustering and bump hunting. *Electronic Journal of Statistics*, 7:499–532, 2013.

José E Chacón, Tarn Duong, and M P Wand. Asymptotics for General Multivariate Kernel Density Derivative Estimators. *Statistica Sinica*, 21(2):807–840, 2011.

K. Chaudhuri, S. Dasgupta, S. Kpotufe, and U. von Luxburg. Consistent procedures for cluster tree estimation and pruning. *IEEE Transactions on Information Theory*, 60(12):7900–7912, 2014.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for the cluster tree. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems - Volume 1*, NIPS'10, pages 343–351, Red Hook, NY, USA, 2010. Curran Associates Inc.

Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete & Computational Geometry*, (1):377–409, dec 1993. doi: 10.1007/BF02573985.

Yen-Chi Chen. A Tutorial on Kernel Density Estimation and Recent Advances. *Biostatistics and Epidemiology*, 1(1):161–187, apr 2017.

Yen Chi Chen, Christopher R. Genovese, and Larry Wasserman. A comprehensive approach to mode clustering. *Electronic Journal of Statistics*, 10(1):210–241, 2016.

Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE transactions on pattern analysis and machine intelligence*, 17(8):790–799, 1995.

Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. Estimating the number of clusters. *Canadian Journal of Statistics*, 28(2):367–382, 2000.

Antonio Cuevas, Manuel Febrero, and Ricardo Fraiman. Cluster analysis: A further approach based on density estimation. *Computational Statistics and Data Analysis*, 36(4):441–459, 2001.

B. Delaunay. Sur la sphère vide. a la mémoire de georges voronoï. *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na*, 6:793–800, 1934.

Justin Eldridge, Mikhail Belkin, and Yusu Wang. Beyond hartigan consistency: Merge distortion metric for hierarchical clustering. volume 40 of *Proceedings of Machine Learning Research*, pages 588–606, Paris, France, 03–06 Jul 2015. PMLR.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231. AAAI Press, 1996.

Chris Fraley and Adrian E. Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631, 2002.

A. L. N. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005. doi: 10.1109/TPAMI.2005.113.

Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory*, 21(1):32–40, 1975.

A. D. Gordon. A Review of Hierarchical Classification. *Journal of the Royal Statistical Society. Series A (General)*, 150(2):119, mar 1987.

Siegfried Graf and Harald Luschgy. *Foundations of Quantization for Probability Distributions*, volume 1730 of *Lecture Notes in Mathematics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000. ISBN 978-3-540-67394-1. doi: 10.1007/BFb0103945.

Siegfried Graf and Harald Luschgy. Rates of Convergence for The Empirical Quantization Error. 30(2):874–897, 2002.

J. A. Hartigan and M. A. Wong. Algorithm AS 136: A K-Means Clustering Algorithm. *Applied Statistics*, 28(1): 100, 1979.

Christian Hennig. Methods for merging gaussian mixture components. *Advances in Data Analysis and Classification 2010 4:1*, 4:3–34, 1 2010.

T. Heskes. Self-organizing maps, vector quantization, and mixture modeling. *IEEE Transactions on Neural Networks*, 12(6):1299–1305, 2001.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, dec 1985.

Jisu Kim, Yen-Chi Chen, Sivaraman Balakrishnan, Alessandro Rinaldo, and Larry Wasserman. Statistical inference for cluster trees. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1839–1847. Curran Associates, Inc., 2016.

G. N. Lance and W. T. Williams. A General Theory of Classificatory Sorting Strategies: 1. Hierarchical Systems. *The Computer Journal*, 9(4):373–380, feb 1967.

Jia Li, Surajit Ray, and Bruce G Lindsay. A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8(8), 2007.

Stuart P. Lloyd. Least Squares Quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

Kenneth Lo, Ryan Remy Brinkman, and Raphael Gottardo. Automated gating of flow cytometry data via robust model-based clustering. In *Cytometry Part A*, volume 73, pages 321–332. Cytometry A, apr 2008.

R. Maitra. Initializing partition-optimization algorithms. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 6(1):144–157, 2009. doi: 10.1109/TCBB.2007.70244.

David M Mason, Wolfgang Polonik, et al. Asymptotic normality of plug-in level set estimates. *The Annals of Applied Probability*, 19(3):1108–1142, 2009.

Giovanna Menardi and Adelchi Azzalini. An advancement in clustering via nonparametric density estimation. *Statistics and Computing*, 24(5):753–767, 2014.

F. Murtagh. A Survey of Recent Advances in Hierarchical Clustering Algorithms. *The Computer Journal*, 26(4): 354–359, nov 1983. doi: 10.1093/comjnl/26.4.354.

Anna D Peterson, Arka P Ghosh, and Ranjan Maitra. Merging k-means with hierarchical clustering for identifying general-shaped groups. *Stat*, 7(1):e172, 2018.

David Pollard. A Central Limit Theorem for k-Means Clustering. *The Annals of Probability*, 10(4):919–926, 1982.

William M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846, dec 1971.

Alessandro Rinaldo, Aarti Singh, Rebecca Nugent, and Larry Wasserman. Stability of density-based clustering. *Journal of Machine Learning Research*, 13:905, 2012.

M. Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9:65–78, 1982.

David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

Luca Scrucca. Identifying connected components in gaussian finite mixture models for clustering. *Computational Statistics & Data Analysis*, 93:5–17, 2016.

Jaehyeok Shin, Alessandro Rinaldo, and Larry Wasserman. Predictive clustering, 2019.

Bernard W Silverman. *Density estimation for statistics and data analysis*, volume 26. CRC press, 1986.

Werner Stuetzle and Rebecca Nugent. A generalized single linkage method for estimating the cluster tree of a density. *Journal of Computational and Graphical Statistics*, 19(2):397–418, 2010.

Maria Tsimidou, Robert Macrae, and Ian Wilson. Authentication of virgin olive oils using principal component analysis of triglyceride and fatty acid profiles: Part 1—classification of greek olive oils. *Food Chemistry*, 25 (3):227 – 239, 1987.

Paxton Turner, Jingbo Liu, and Philippe Rigollet. A statistical perspective on coreset density estimation, 2020.

G Voronoi. Recherches sur les parallélоèdres primitives. *J. reine angew. Math*, 134:198–287, 1908.

Marek Walesiak and Andrzej Dudek. The choice of variable normalization method in cluster analysis. In Khalid S. Soliman, editor, *Education Excellence and Innovation Management: A 2025 Vision to Sustain Economic Development During Global Challenges*, pages 325–340. International Business Information Management Association (IBIMA), 2020. ISBN 978-0-9998551-4-1.

Matt P Wand and M Chris Jones. Multivariate plug-in bandwidth selection. *Computational Statistics*, 9(2): 97–116, 1994.

Larry Wasserman. *All of Nonparametric Statistics*. Springer New York, 2006. doi: 10.1007/0-387-30623-4.

Roger Weber, Hans-Jörg Schek, and Stephen Blott. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *Proceedings of the 24rd International Conference on Very Large Data Bases*, VLDB '98, pages 194–205, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 1558605665.

Appendix

A Tube Density

While FD is conceptually appealing, the characterization of the face between two Voronoi cells could be challenging since the shapes of the boundaries can be irregular. Here we propose a measure similar to the Face density measure but has a predefined regular shape. For a point x, we define the *Disk Area* centered at x with radius R and normal direction ν (see Figure 7 for an illustration) as

$$\mathsf{Disk}(x, R, \nu) = \{y : ||x - y|| \leq R, (x - y)^T \nu = 0\} \tag{10}$$

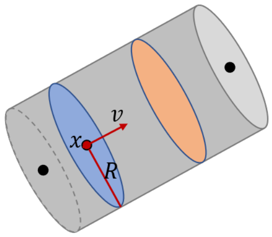


Figure 7: The disk area centered at x with a radius R and a direction ν.

To measure the similarity between knots c_j and c_ℓ, we examine the integrated density within the disk areas along the central line. In more details, the central line can be expressed as $\{c_j + t(c_\ell - c_j) : t \in [0, 1]\}$, and any point on the central line can be written as $c_j + t(c_\ell - c_j)$ for some t. For a point $c_j + t(c_\ell - c_j)$, we define the integrated density in the disk region (called *Disk Density*) as

$$\mathsf{pDisk}_{j\ell,R}(t) = \mathbb{P}\left(\mathsf{Disk}(c_j + t(c_\ell - c_j), R, c_\ell - c_j)\right) = \int_{\mathsf{Disk}(c_j + t(c_\ell - c_j), R, c_\ell - c_j)} p(x)dx. \tag{11}$$

The *Tube Density (TD)* measures the similarity between c_j and c_ℓ as the minimal disk density along the central line, i.e.,

$$S^{TD}_{j\ell} = \inf_{t \in [0,1]} \mathsf{pDisk}_{j\ell,R}(t) \tag{12}$$

In other words, with given c_j, c_ℓ, we survey all Disk Density along the central line and retrieve the infimum as the similarity measure between two knots.

In this work, we set R based on the root mean squared distances within each Voronoi cell. Specifically, for knot c_j and the corresponding Voronoi cell $\mathbb{C}_j$, we calculate

$$R_j = \sqrt{\frac{1}{|\mathbb{C}_j| - 1} \sum_{X_\ell \in \mathbb{C}_j} \|X_\ell - c_j\|^2} \tag{13}$$

where $|\mathbb{C}_j|$ denotes the size of set $\mathbb{C}_j$. With the uniform radius paradigm where the radius is the same for all pairs of knots, we set $R = \frac{1}{k}\sum_{j=1}^{k} R_j$. Our empirical studies show that this rule leads to good clustering performances.

Note that the radius may also be chosen adaptively for each pair: we set the disk radius at c_j to be R_j for all knots and set the disk radius along the edge to be the linear interpolation of the radii at the two connected knots. The comparison between the uniform and adaptive R is presented in Appendix D.6, and similar clustering performance is observed for the two approaches. Hence we use uniform R by default for simplicity.

Similar to the FD, we estimate the TD by a projected KDE. Let $\Pi_{j\ell}(x)$ be the projection of a point x on the line through c_j, c_ℓ. We first estimate the pDisk via

$$\widehat{\mathsf{pDisk}}_{j\ell,R}(t) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{\Pi_{j\ell}(X_i) - c_j - t(c_\ell - c_j)}{h}\right) I(||X_i - \Pi_{j\ell}(X_i)|| \leq R) \tag{14}$$

and then estimate the TD as

$$\hat{S}_{j\ell}^{TD} = \inf_{t\in[0,1]} \widehat{\mathsf{pDisk}}_{j\ell,R}(t). \tag{15}$$

where the infimum is approximated by grid search.

The estimations of the FD and the TD involve the use of the projected kernel density estimation, and we discuss the choices of kernel and the bandwidth selections for kernel density estimations in Appendix D.3. By default, we use the Gaussian kernel with the normal scale bandwidth selector (NS) [Chacón et al., 2011] for the best empirical results.

B Computational Complexity

Knots construction. The first step of skeleton clustering is choosing knots, and in this work we take overfitting k-means as the default method. The k-means algorithm of Hartigan and Wong [Hartigan and Wong, 1979] has time complexity $O(ndkI)$, where n is the number of points, d is the dimension of the data, k is the number of clusters for k-means, and I is the number of iterations needed for convergence. When using overfitting k-means to chooses knots, the reference rule is $k = \sqrt{n}$, and hence the complexity is $O(n^{3/2}dI)$. This is a time consuming step of our clustering framework, and the complexity increases linearly with d. Therefore, preprocessing the data with dimension reduction techniques or using subject knowledge to choose knots can be helpful to speed up this process.

Edges construction. For the edge construction step, we approximate the Delaunay Triangulation with $\hat{DT}(\mathcal{C})$ by looking at the 2-NN neighborhoods (the Voronoi Density regions in 3.1). Hence the main computational task for our edge construction step is the 2-nearest knot search. We used the k-d tree algorithm for this purpose, which gives the worst-case complexity of $O(ndk^{(1-1/d)})$. Notably, the computation complexity at this step is at the worst linear in d, which is a much better rate than computing the exact Delaunay Triangulation (exponential dependence on d), and our empirical studies have illustrated the effectiveness of such approximation.

Edge weight construction: VD. Next, we consider the computation complexity of the different edge weights measurements. For the VD, its numerator can be computed directly from the 2-NN search when constructing the edges and hence no additional computation is needed. The denominators are pairwise distances between knots and can be computed with the worst-case complexity of $O(dk^2)$ because the number of nonzero edges is less than $\frac{k(k-1)}{2}$. With $k = \sqrt{n}$, we have the total time complexity of computing the VD to be $O(nd)$.

Edge weight construction: FD. For the Face density, we calculate the projected KDE at the middle point for each pair of neighboring Voronoi cells. The projection of one data point onto one central line can be done by matrix multiplication with complexity $O(d)$. Recall that we only use data points in local Voronoi cells for FD calculation, and the local sample size would be at $n_{loc} = O(\sqrt{n})$ under the reference rule $k = [\sqrt{n}]$. Together it takes $O(d\sqrt{n})$ to calculate the projected data for one edge. With the projected data, KDE calculation has a time complexity $O(c \log c)$ where $c = \max_{j\neq\ell}\{n_j + n_\ell\}$ for any pair of knot indexes j, ℓ. Again we have $c = O(n/k) = O(\sqrt{n})$ under the previously mentioned conditions. We need to do KDE for each edge in the skeleton, which gives the overall time complexity of FD weights to $O(k^2 d\sqrt{n} + k^2 c \log c) = O(n^{3/2}d + n^{3/2}\log n)$.

Edge weight construction: TD. For Tube density, we similarly perform a projected KDE for each edge. Let η be the maximum number of points in a tube region $\eta = \max_{j,\ell} |\{X_i : \|\Pi_{j\ell}(X_i) - X_i\| \leq R\}|$, the data projection again takes $O(\eta d)$ complexity. Suppose the minimum density is obtained by a grid search with m grid points, the KDE step takes a total of $O(m\eta \log \eta)$ for one edge. To compute the whole edge weights matrix with $k = \sqrt{n}$, we have the complexity to be $O(n\eta d + nm\eta \log \eta)$. Under conditions where the tube regions for TD estimations is also of size $\eta = O(n/k) = O(\sqrt{k})$, we have the overall complexity for VD weights calculation to be $O(k^2 d\sqrt{n} + k^2 c \log c) = O(n^{3/2}d + mn^{3/2} \log n)$, which is larger than that for FD due to the grid search for minimum density.

Knots segmentation. In this work, we segment the learned weighted skeleton using hierarchical clustering. With links that can be updated by Lance-Williams update [Lance and Williams, 1967] and satisfies the reducibility condition [Gordon, 1987], hierarchical clustering can be carried out with computation complexity $O(N^2)$, where N is the number of points to start the algorithm with [Murtagh, 1983]. For our empirical results we favored single linkage and average linkage, and both satisfy the requirements for efficient hierarchical clustering algorithm. We perform hierarchical clustering on the $k = \sqrt{n}$ knots, and hence the computation complexity for segmenting the skeleton structure is $O(k^2) = O(n)$.

C Proofs

C.1 Voronoi Density Consistency

We restate the assumption:

(B1) There exists a constant c_0 such that the minimal knot size $\min_{(j,\ell)\in E} \mathbb{P}(A_{j\ell}) \geq \frac{c_0}{k}$ and $\min_{(j,\ell)\in E} \|c_j - c_\ell\| \geq \frac{c_0}{k^{1/d}}$, where $A_{j\ell}$ is the 2-NN region of knots c_j, c_ℓ as defined in Equation 2.

of Theorem 1. For given knots c_j, c_ℓ, the distance $||c_j - c_\ell||$ is also given. We denote the numerator of $S^{VD}_{j\ell}$ as

$$p_{j\ell} = \mathbb{P}(A_{j\ell}) = \mathbb{E}I(X_i : d(X_i, c_m) > max\{d(X_i, c_j), d(X_i, c_\ell), \forall m \neq j, l\})$$

and note that the numerator of $\hat{S}^{VD}_{j\ell}$ is

$$\hat{P}_n(A_{j\ell}) = \frac{1}{n}\sum_{i=1}^{n} I(X_i : d(X_i, c_m) > max\{d(X_i, c_j), d(X_i, c_\ell), \forall m \neq j, l\}),$$

which is a sum of binary variables and has variance $\sigma^2_{j\ell} = \frac{p_{j\ell}(1-p_{j\ell})}{n}$. By the Chebyshev's inequality,

$$\left|\hat{P}_n(A_{j\ell}) - p_{j\ell}\right| = O_p(\sigma^{1/2}_{j\ell}) = O_p\left(\left[\frac{p_{j\ell}(1-p_{j\ell})}{n}\right]^{1/2}\right)$$

Note that the region $A_{j\ell}$ is changing with respect to k. The ratio is then

$$\begin{aligned}\left|\frac{\hat{S}^{VD}_{j\ell}}{S^{VD}_{j\ell}} - 1\right| = \left|\frac{\hat{P}_n(A_{j\ell})}{\mathbb{P}(A_{j\ell})} - 1\right| &= \frac{1}{p_{j\ell}} O_p\left(\left[\frac{p_{j\ell}(1-p_{j\ell})}{n}\right]^{1/2}\right)\\ &= O_p\left(\left[\frac{(1-p_{j\ell})}{np_{j\ell}}\right]^{1/2}\right) = O_p\left(\left[\frac{(1-c_0/k)}{nc_0/k}\right]^{1/2}\right) = O_p\left(\left(\frac{k}{n}\right)^{1/2}\right)\end{aligned}$$

by assumption (B1) that $\min_{(j,\ell)\in E} \mathbb{P}(A_{j\ell}) \geq \frac{c_0}{k}$, which completes the proof for Equation 7.

To get the uniform bound, we first start with the concentration bound. Note that $\big(I(X_i \in A_{j\ell}) - p_{j\ell}\big)$ has zero mean and $|I(X_i \in A_{j\ell}) - p_{j\ell}| \leq 1$. Hence by Bernstein inequalities we have

$$\begin{aligned}
\mathbb{P}\left\{\left|\frac{\hat{P}_n(A_{j\ell})}{p_{j\ell}} - 1\right| > \varepsilon\right\} &= \mathbb{P}\left\{\left|\hat{P}_n(A_{j\ell}) - p_{j\ell}\right| > \varepsilon p_{j\ell}\right\} \\
&= \mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^{n} I(X_i \in A_{j\ell}) - p_{j\ell}\right| > \varepsilon p_{j\ell}\right\} \\
&= 2\mathbb{P}\left\{\sum_{i=1}^{n} (I(X_i \in A_{j\ell}) - p_{j\ell}) > n\varepsilon p_{j\ell}\right\} \\
&\leq 2\exp\left\{-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n^2}{\sum_{i=1}^{n} \mathbb{E}\left[(I(X_i \in A_{j\ell}) - p_{j\ell})^2\right] + \frac{1}{3}\varepsilon p_{j\ell} n}\right\} \\
&= 2\exp\left\{-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n^2}{np_{j\ell}(1 - p_{j\ell}) + \frac{1}{3}\varepsilon p_{j\ell} n}\right\} \\
&= 2\exp\left\{-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n}{p_{j\ell}(1 - p_{j\ell}) + \frac{1}{3}\varepsilon p_{j\ell}}\right\}
\end{aligned}$$

Note that plugging in the $p_{j\ell} = \Omega\left(\frac{1}{k}\right)$ rate to above concentration bound we can recover the $O_p\left(\sqrt{\frac{k}{n}}\right)$ rate in Equation 7. Then by union bound we have

$$\begin{aligned}
\mathbb{P}\left\{\max_{(j,\ell)\in\mathcal{S}} |\hat{S}_{j\ell}/S_{j\ell} - 1| > \varepsilon\right\} &\leq \mathbb{P}\left\{\max_{j,\ell} |\hat{S}_{j\ell}/S_{j\ell} - 1| > \varepsilon\right\} \\
&\leq \sum_{j,\ell} \mathbb{P}\left\{|\hat{S}_{j\ell}/S_{j\ell} - 1| > \varepsilon\right\} \\
&\leq \frac{k(k-1)}{2} \max_{j,\ell} \mathbb{P}\left\{\left|\frac{\hat{P}_n(A_{j\ell})}{p_{j\ell}} - 1\right| > \varepsilon\right\} \\
&\leq k(k-1) \max_{j,\ell} \left\{\exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{j\ell}^2 n}{p_{j\ell}(1 - p_{j\ell}) + \frac{1}{3}\varepsilon p_{j\ell}}\right)\right\} \\
&\leq k(k-1)\exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1 - p_{min}) + \frac{1}{3}\varepsilon}\right)
\end{aligned}$$

where $p_{min} = \min_{j\ell} p_{j\ell}$. Therefore we can derive the uniform error bound that

$$\max_{j,\ell} \left|\frac{\hat{S}_{j\ell}^{VD}}{S_{j\ell}^{VD}} - 1\right| = O_p\left(\sqrt{\frac{k}{n}}\log k\right),$$

when $n \to \infty, k \to \infty, \frac{n}{k} \to \infty$.

□

Proof. of Theorem 2 (Performance guarantee for Voronoi density) We note that, assuming (P1),

$$\begin{aligned}
\mathbb{P}\left\{ARI(\mathcal{L}^*, \hat{\mathcal{L}}) < 1\right\} &\leq \mathbb{P}\left\{\text{there exists at least one wrongly cut edge}\right\} \\
&= \mathbb{P}\left\{\max_{(j,\ell)\in\mathcal{S}} |\hat{S}_{j\ell}/S_{j\ell} - 1| > \varepsilon\right\} \\
&\leq k(k-1)\exp\left(-\frac{\frac{1}{2}\varepsilon^2 p_{min} n}{(1 - p_{min}) + \frac{1}{3}\varepsilon}\right)
\end{aligned}$$

□

by the uniform bound derived above.

D Additional Data Analysis

D.1 Performance with Different Number of Knots

We analyze how the number of knots would affect the performance of the skeleton clustering. We empirically test the effect of the number of knots, k, on the final clustering performance on Yinyang data with dimensions $10, 100, 500$ and 1000. For each dimension, we simulated the Yinyang data 100 times, and for each simulated data we carried out the default skeleton clustering procedure with single linkage and different k (other steps the same as in Section 5.1). Figure 8 displays the median adjusted Rand index given by each method across different k, where the reference rule with $k = 57$ is marked by the vertical dash line. We see that as long as k is sufficiently large, skeleton clustering works well.

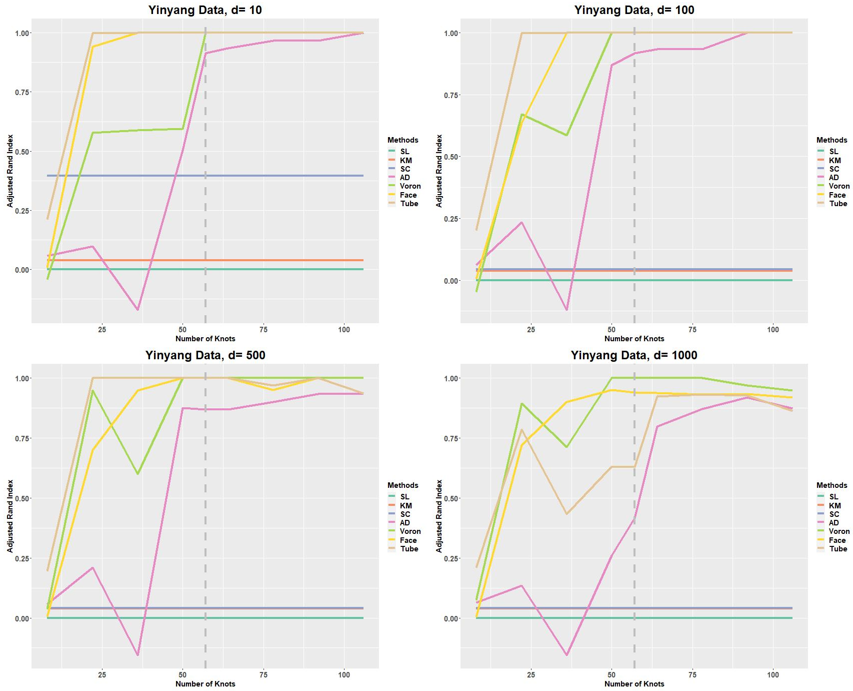


Figure 8: Adjusted Rand indexes of different clustering methods against different number of knots on 100 simulated Yinyang data.

D.2 Self-Organizing Map

The Self-Organizing Map (SOM) is another popular prototype clustering method and can be used as an alternative to k-means clustering in finding knots. Thus, here we conduct a simple experiment to examine the performance of using SOM to find knots. We examine the performance using Yingyang data with $d = 10$ to $d = 1000$. The identical procedure as in Section 5.1 is applied except that the knots are now detected by the SOM rather than overfitting k-means. The total number of grid points in the SOM is the total number of knots we obtain and, to be comparable to k-means with $k = \sqrt{n}$ knots, we used $\lceil n^{1/4} \rceil$ breaks for each dimension of the SOM grid, giving a total of $\lceil n^{1/4} \rceil^2$ initial grid points. However, the SOM may return knots with very tiny sample size, on which the density-aided similarity measures cannot be calculated. Therefore, we remove knots with less than 3 data points and use the remaining ones for skeleton construction.

Figure 9 summarizes the result. The top left panel shows the knots from the SOM (after post-processing), which are located around the main data structures and are representative to the original data as well. The dendrogram shows the cluster structure of the SOM knots using Voronoi density on one 100-dimensional Yinyang data. In the bottom row, we display the adjusted Rand indices from the clustering methods. Compared to the results of Figure 5, the adjusted Rand indices given by the skeleton clustering with SOM knots are similarly good when the dimension is not so high ($d = 10$ and 100). But when the data dimension becomes high ($d = 500, 1000$), knots constructed by SOM lead to worse clustering results. Therefore, overfitting k-means is favored in this work. Another limitation of SOM is that we need to perform some post-processing to remove tiny knots; in the case of k-means, we do not need such procedure.

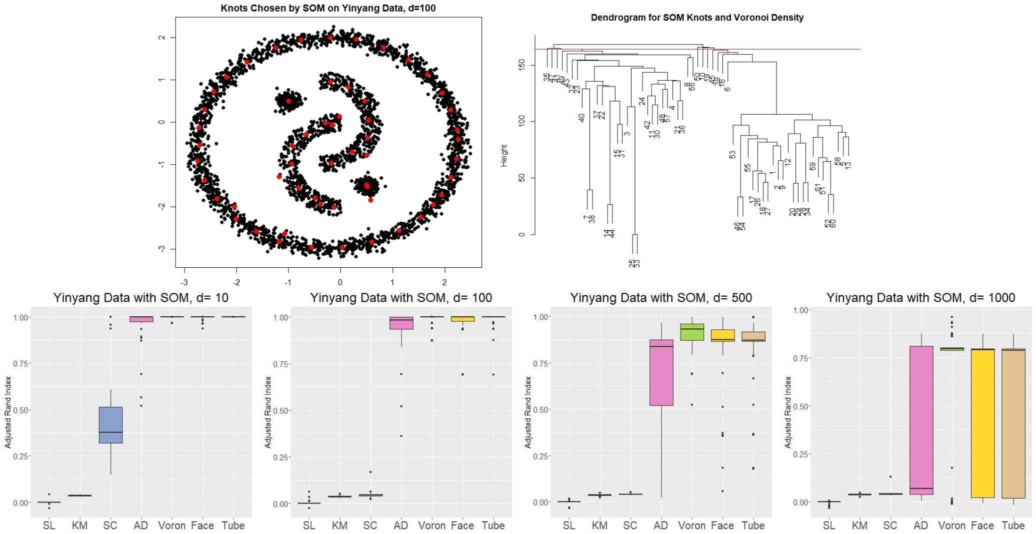


Figure 9: Adjusted Rand indexes using SOM for knots selection on Yinyang data.

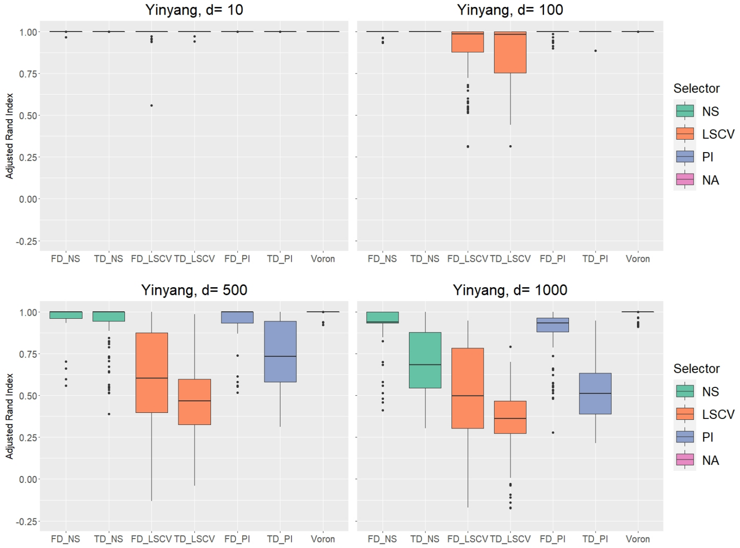


Figure 10: Performance of skeleton clustering on Yinyang data $d = 10, 100, 500, 1000$ with Face and Tube density by different bandwidth selectors. Voronoi density result is included for comparison.

D.3 Bandwidth Selection Yinyang Data

The estimations of the FD and the TD involve the use of the projected kernel density estimation, for which the type of kernel and the bandwidth need to be specified. Similar to the usual KDE, the kernel function does not affect the final performance much, so by default we use the Gaussian kernel in all of our empirical studies. It is worth noting that using the uniform kernel can save some computation since it has compact support, but empirically we find using the Gaussian kernel leads to better final clustering results. In what follows, we focus on the bandwidth selection.

It is known that the bandwidth is a pivotal parameter that can significantly affect the estimation result of a kernel density estimator. In Figure 10, we conduct a simulation using the Yinyang data with different dimensions of noisy Gaussian variables (see Section 5.1 for more details) and compare the performance of three common bandwidth selectors: the normal scale bandwidth (NS) [Chacón et al., 2011], the least-squared cross-validation (LSCV) [Bowman, 1984, Rudemo, 1982], and the plug-in approach (PI) [Wand and Jones, 1994]. Each edge is allowed to have its own bandwidth. Voronoi density performance results are also included for comparison. We found that the NS performs reliably well while the others may have unstable performance. A similar comparison of the bandwidth selectors on another dataset is presented in Appendix D.4 and the NS also performs relatively better than the other bandwidth selectors.. As a result, we recommend using the NS as the default bandwidth selector. Additionally, since the density estimations are all 1-dimensional, in practice it is possible to examine the estimated density to assess the degree of oversmoothing or undersmoothing and manually adjust the bandwidth.

In addition to different bandwidth selectors, we also study how the bandwidth should depend on the sample size for clustering purpose. In 1-dimensional data, the normal scale bandwidth agrees with Silverman's rule of thumb [Silverman, 1986] giving the bandwidth as $h = \frac{4}{3}^{1/5}\hat{\sigma} n_{loc}^{-1/5}$, where $\hat{\sigma}$ is the standard deviation of the sample used in the edge weight calculation, and n_{loc} the number of sample points used. Empirically we tested the clustering performance with FD and TD calculated under bandwidth with rates on n_{loc} from $-1/3$ to $-1/10$ (see Appendix D.5). We found that the clustering performance with FD and TD generally stays stable with varying bandwidth rates, although a larger bandwidth (slower rate than $O(n_{loc}^{-1/5})$) may give better clustering results with TD when the dimension of the data is high.

D.4 Bandwidth Selection with Mix Mickey

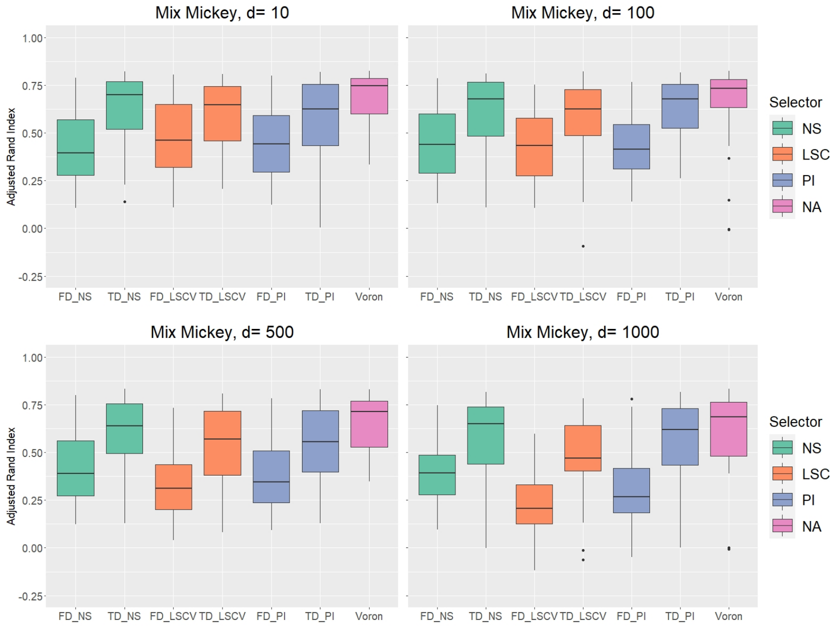


Figure 11: Performance of skeleton clustering on Mix Mickey data $d = 10, 100, 500, 1000$ with Face and Tube density by different bandwidth selectors. Voronoi density result is included for comparison.

We present additional results comparing different bandwidth selectors on the Mix Mickey dataset generated the same way as in Appendix E.2. We use average linkage for all the included skeleton clustering approaches. The results are presented in Figure 11. The selectors have similar performances on this Mix Mickey dataset, but NS again seems to perform better with larger dimensions, which corroborates our default choice of using NS for bandwidth.

D.5 Performance under Different Bandwidth Rate

In this section we present empirical results on how changing the bandwidth rate affects the performance of clustering. We consider the Yinyang data in Section 5.1 with $d = 10, 100, 500, 1000$. We compare the Face and Tube density where the bandwidth is selected by Silverman's rule of thumb with different rates, ranging from $n_{loc}^{-1/3}$ to $n_{loc}^{-1/10}$. Note that the original Silverman's rule of thumb will be at rate $n_{loc}^{-1/5}$. We repeat the experiment 100 times and record the adjust Rand index in Figure 12.

When the dimension is low (top panels), all bandwidth within this range works well. When the dimension is large (bottom panels), a slower rate (larger bandwidth) seems to be showing a better performance for the TD. Interestingly, the face density yields a robust result across different rates of bandwidth. Note that for the TD, the univariate density estimation theory suggests the choice at rate $h \asymp n_{loc}^{-1/5}$ is optimal for estimation in large d, the same rate may not lead to a the optimal clustering performance. Figure 12 bottom-right panel suggests that the choice $h \asymp n_{loc}^{-1/10}$ may have a better clustering performance in this case.

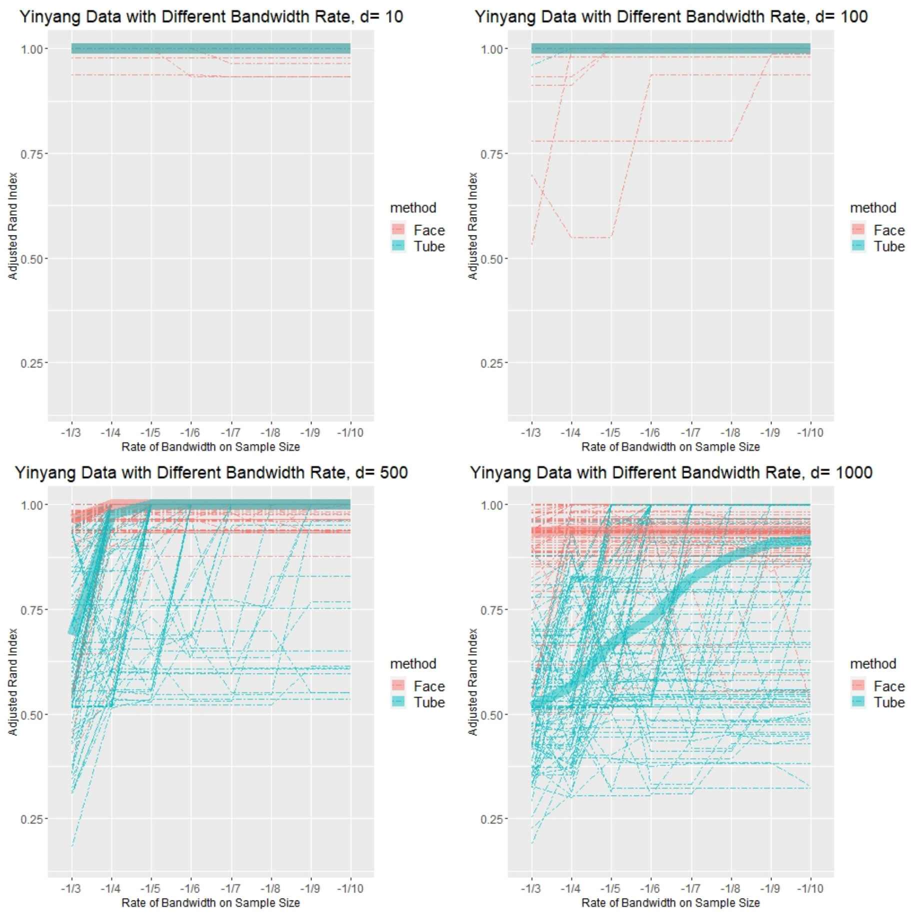


Figure 12: Adjusted Rand indexes of skeleton clustering with Face and Tube density under different bandwidth rate on 100 simulated Yinyang datasets. The thick lines indicate the median adjusted Rand index of a given method.

D.6 Adaptive Radius for Tube Density

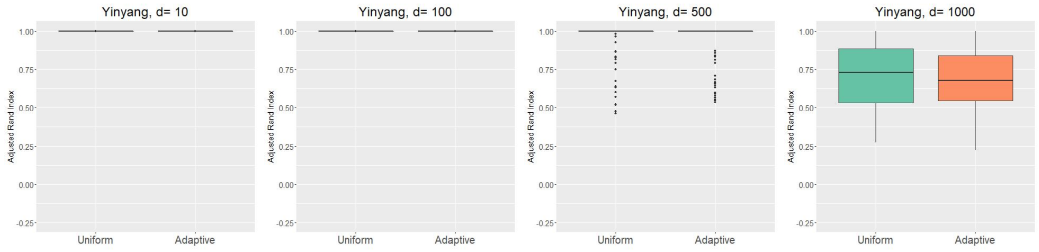


Figure 13: Comparison of radius choices on Yinyang data with dimensions 10, 100, 500, 1000.

We compare the clustering performance of Tube density when using fixed radius and that when using adaptive radius as described in Section A. The data is the same Yinyang data in Section 5.1 and the results are presented in Figure 13. The two approaches (adaptive and fixed radius) have a similar performance.

D.7 Higher Standard Deviations for Noisy Dimensions

We investigate how does changing the noise level of the added noisy dimensions of our simulation examples change the clustering performance. Here we simulate Yinyang data with different standard deviations of the added dimensions. We apply the same analysis procedure as in Section 5.1 is applied. The adjusted Rand indexes of each clustering methods on 100 simulated datasets with under setting are presented in Figure 14.

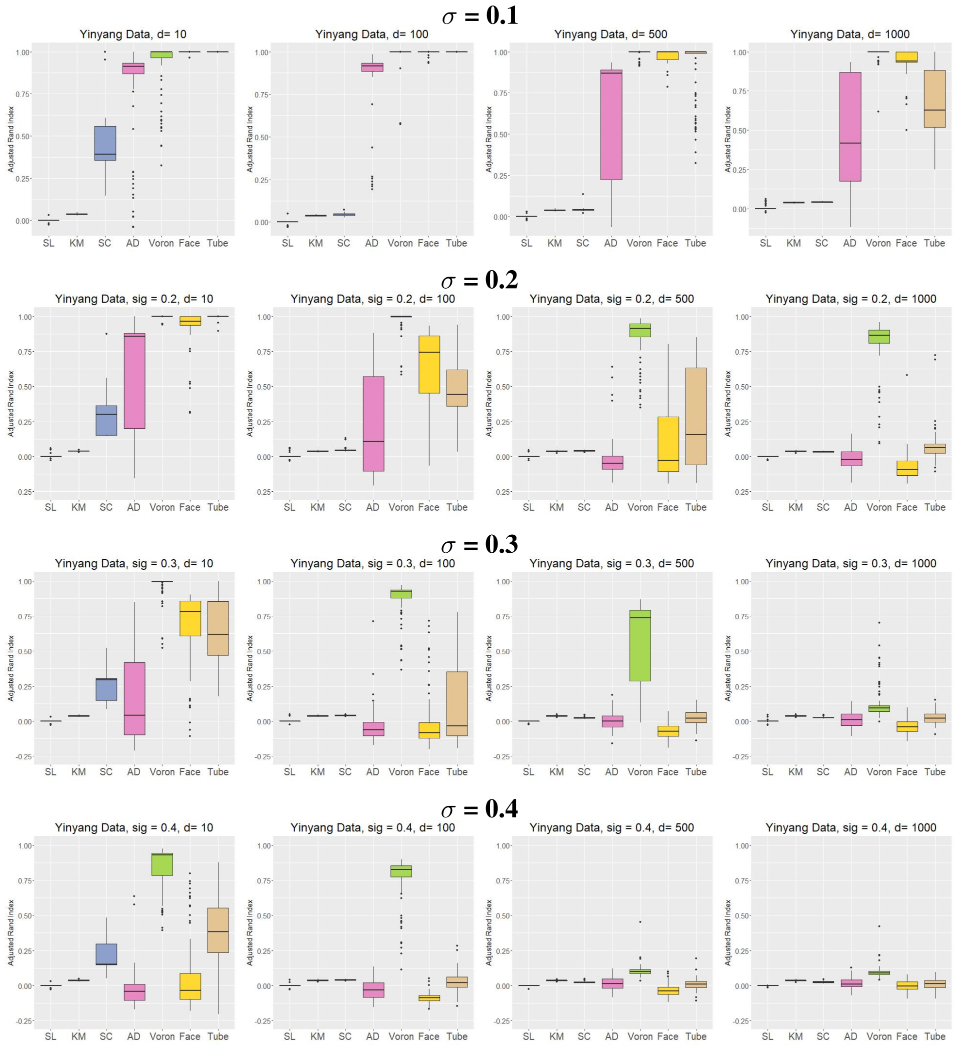


Figure 14: Adjusted Rand index performance of clustering methods on Yinyang data with different standard deviation for added dimensions.

We observe that increasing the standard deviation of the noisy dimensions (noise level) has a stronger impact than adding more noisy variables. For example, increasing $\sigma = 0.1 \rightarrow 0.2$ scales the standard deviation by a factor of 2 (scales the variance 4 times), but the clustering performance with $\sigma = 0.2, d = 100$ is worse than that with $\sigma = 0.1, d = 500$. However, we still observe that the skeleton clustering with Voronoi density similarity measure can give good clustering performance even under the setting with $\sigma = 0.4$ and $d = 100$.

D.8 Mix Mickey with GMM

We compare the performance of Gaussian Mixture Models (GMMs) to our methods using the Mix Mickey data same as in Section E.2. Unfortunately, the GMM method from `clusterR` package in R cannot work with dimension 500 and 1000 case because of too much noisy dimensions, so we only compare the case of dimension 10 and 100. For the skeleton clustering, we use average linkage for the segmentation step the same as in Section E.2. Because this data is generated from 3-GMM and we fit the GMM with 3 components, the GMM naturally has the best performance. However, our proposed approaches may achieve a comparable performance to the GMM and are capable of handling high dimensional data ($d = 500, 1000$).

D.9 Graphical Representation of GvHD Data Clusters

We visualize the skeleton structure of the clusters identified on the GvHD dataset in Section F.1. These graph representations are generated by the `igraph` package in R. Cluster 6 only has 1 knot with 17 corresponding data

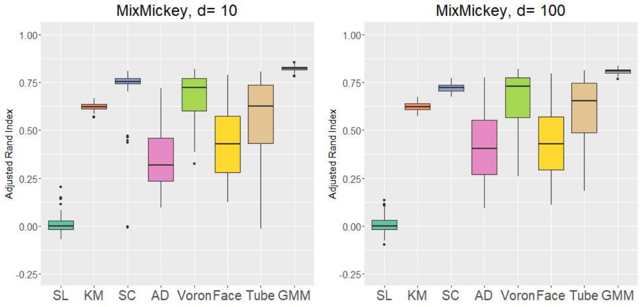
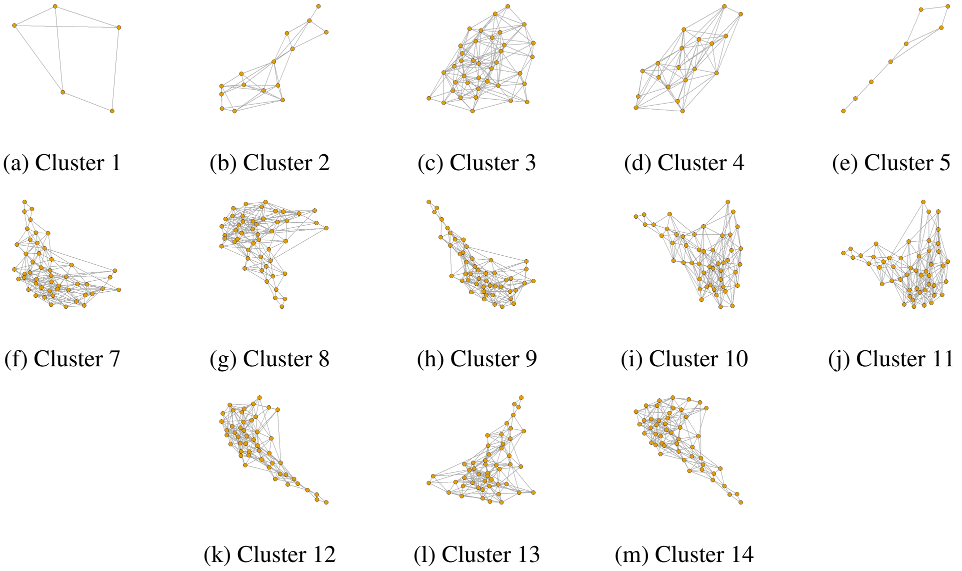


Figure 15: Comparison of clustering methods on Mix Mickey data $d = 10, 100$ with GMM included.



(a) Cluster 1 (b) Cluster 2 (c) Cluster 3 (d) Cluster 4 (e) Cluster 5

(f) Cluster 7 (g) Cluster 8 (h) Cluster 9 (i) Cluster 10 (j) Cluster 11

(k) Cluster 12 (l) Cluster 13 (m) Cluster 14

Figure 16: Skeleton structures of the clusters identified for the GvHD dataset in Section F.1

points and is hence omitted in Figure 16. We observe that most clusters display a hammer-like structure, which is non-spherical and not favorable for some classical clustering methods. Only Cluster 3 has a spherical shape in this data.

E Additional Simulated Data Examples

E.1 Mickey Data

The simulated Mickey data is an intrinsically 2-dimensional data consists of one large circular region with 1000 data points and two small circular regions each with 100 data points. As a result, the structures have unbalanced sizes. The total sample size is $n = 1200$ and we choose the number of knots to be $k = [\sqrt{1200}] = 35$. We include additional variables with random Gaussian noises to make it a high dimensional data ($d = 10, 100, 500, 1000$) the same way as in Section 5.1. The left panel of Figure 17 shows the scatter plot of the first two dimensions.

We perform the same comparisons as done on the Yinyang data with the true number of components $S = 3$ provided to all the clustering algorithms, and the results are displayed in Figure 18. All methods perform well when d is small but starting at $d = 100$, traditional methods fail to recover the underlying clusters. On the other hand, all methods in the skeleton clustering framework work well even when $d = 1000$.

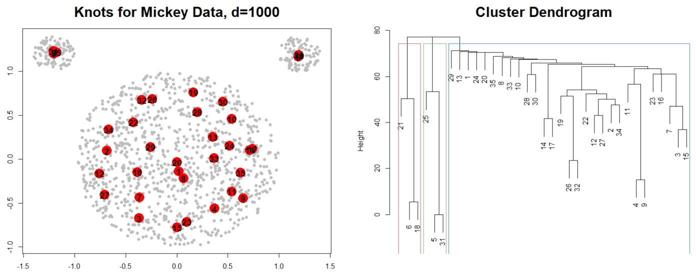


Figure 17: An illustration of the analysis of the Mickey data with dimension 100.

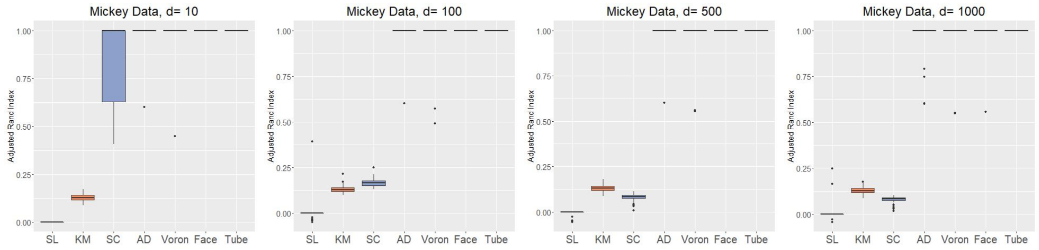


Figure 18: Comparison of adjusted Rand index using different similarity measures on Mickey data with dimensions 10, 100, 500, 1000.

E.2 Mix Mickey Data

The well-separated structures in the Yingyang data may provide advantages to the single linkage. To investigate the effect of linkage criteria on the overlapping clusters, we consider a three-Gaussian mixture model in 2D case that we call it the Mix Mickey data. The large cluster is centered at $(0, 0)$ with the covariance matrix being a diagonal matrix of 2 and has 2000 points. The two smaller clusters are centered at $(3, 3)$ and $(-3, 3)$ respectively, and both have a covariance matrix being a diagonal matrix of 1, and each has 600 points. Random Gaussian variables are added to make the data $d = 10, 100, 500, 1000$ dimensions via the same way we generate the Yinyang data. Figure 19 presents a scatter plot of the first two dimensions; the three clusters have a substantial amount of overlap so that it is difficult for clustering methods to separate them into three distinct clusters. The results under the same linkages analysis pipeline are shown in Figure 20.

Remark 1. GMM can be favored in this data example but is unstable and cannot work with too many noisy dimensions. We present some comparisons including GMM in Appendix D.8.

We observe that average linkage gives good performance at $S = 3$ (the true number of clusters) and single linkage fails to give a satisfying performance under this scenario, giving non-informative clusters at low S (only extracting small clusters) and too fragmented clusters at high S. The average linkage is a criterion that tends to create spherical clusters with similar sizes and hence is better suited for this simulated data. Thus, our experiment shows that, for data containing overlapping clusters with roughly spherical shapes, the average linkage criterion in the knots segmentation step is preferred.

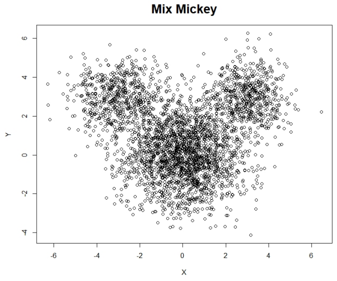


Figure 19: First two dimensions of Mix Mickey data.

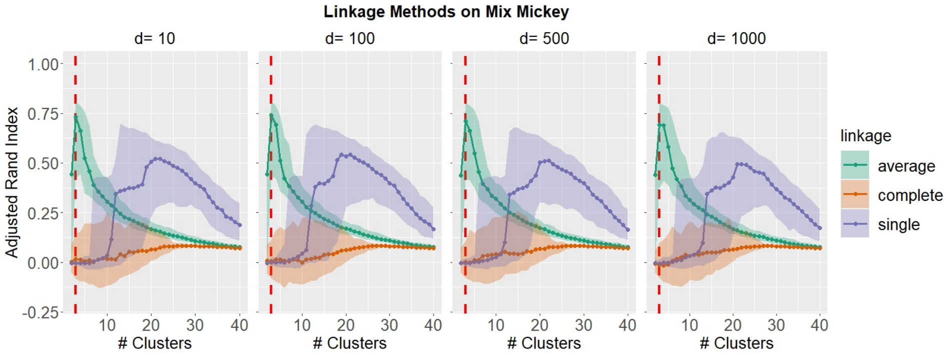


Figure 20: Clustering results with different linkage methods across different numbers of final clusters on Mix Mickey data. The vertical red dashed line indicates the true number of 3 clusters.

E.3 Manifold Mixture Data

In the Yinyang data and the Mix Mickey data experiments, the underlying components are all two-dimensional structures. Here we consider the data composed of structures of different intrinsic dimensions called the manifold mixture data. The simulated manifold mixture data, as illustrated in the left panel of Figure 21, consists of a 2-dimensional plane with 2000 data points, a 3-dimensional Gaussian cluster with 400 data points, and an essentially 1-dimensional ring shape with 800 data points. There are a total of 3200 observations and we choose $k = [\sqrt{3200}] = 57$ knots. Similar to the other two simulations, we include Gaussian noise variables to make the data high-dimensional ($d = 10, 100, 500, 1000$) and make comparisons between the same set of clustering methods. The true number of components $S = 3$ is provided to all the clustering algorithms.

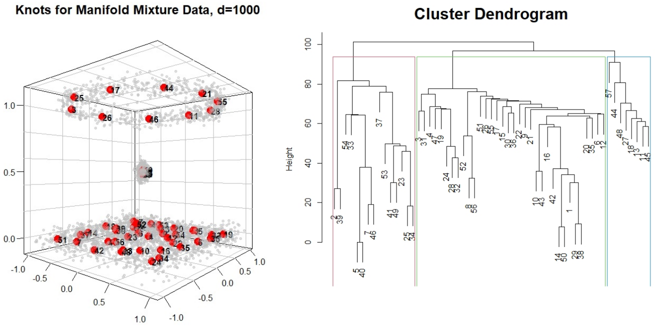


Figure 21: Results on Manifold Mixture data with dimension 100.

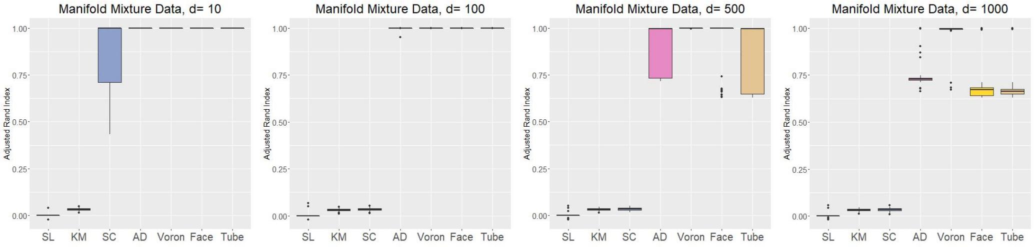


Figure 22: Comparison of adjusted Rand index using different similarity measures on Manifold Mixture data with dimensions 10, 100, 500, 1000.

Figure 22 summarizes the performance of each method. Traditional methods (SL, KM, and SC) do not perform well when $d > 10$ while all methods of skeleton clustering perform very well when $d \leq 500$. Notably, the skeleton clustering with VD still has a perfect performance even when $d = 1000$, whereas skeleton clustering based on other similarity measures gives satisfying results.

E.4 Ring Data

The ring data is constructed by a mixture distribution such that with a probability of $\frac{1}{6}$ we sample from the ring structure and with a probability of $\frac{5}{6}$ we sample from the central part. The ring structure is generated by a uniform distribution over the ring $\{(x_1, x_2) : x_1^2 + x_2^2 = 1\}$ and is corrupted with an additive Gaussian noise $N(0, 0.2^2\mathbf{I}_2)$. The central part is simply a Gaussian $N(0, 0.2^2\mathbf{I}_2)$. We generate a total of $n = 1200$ points from the above mixture and add the high dimensional noise with the same procedure as in Section 5. The same skeleton clustering approached are applied as well as the classical approaches, with the final number of clusters chosen to be 2. The result is displayed in Figure 24. Again, the density-based skeleton clustering methods work well even when the dimension is large.

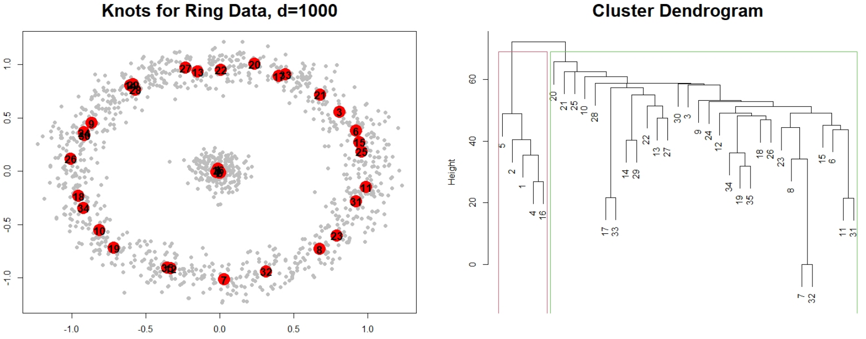


Figure 23: Results on Ring data with dimension 1000.

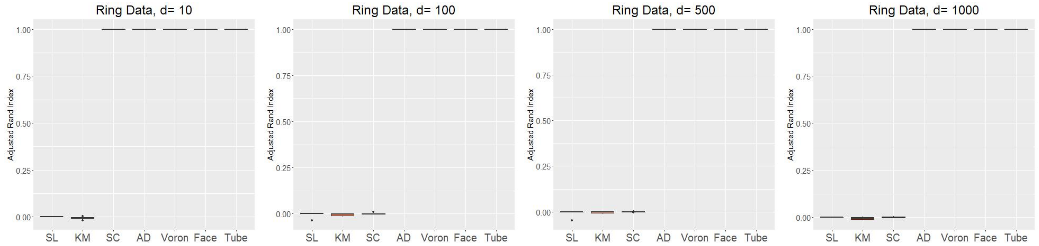


Figure 24: Comparison of the rand index using different similarity measures on Ring data with dimensions 10, 100, 500, 1000. Medium of 100 repetitions.

F Real Data Examples

F.1 GvHD Data

In this section, we apply skeleton clustering to one real data example: the graft-versus-host disease (GvHD) data [Brinkman et al., 2007]. Additionally, we analyze the Zipcode data [Stuetzle and Nugent, 2010] in Appendix F.2 and the Olive Oil data [Tsimidou et al., 1987] in Appendix F.3.

GvHD is a significant problem in the field of allogeneic blood and marrow transplantation which occurs when allogeneic hematopoietic stem cell transplant recipients when donor-immune cells in the graft attack the tissues of the recipient. The data include samples from a patient with GvHD containing $n_1 = 9083$ observations and samples from a control patient with $n_2 = 6809$ observations. Both samples include four biomarker variables, CD4, CD8β, CD3, and CD8. Previous studies [Lo et al., 2008, Baudry et al., 2010] have identified the presence of high values in CD3, CD4, CD8β cell sub-populations as a significant characteristic in the GvHD positive sample and a major objective of our analysis is to rediscovery this region with the proposed skeleton clustering methods. In addition, our skeleton clustering procedure shows more information and leads to a novel two-sample test.

The two samples are plotted in the left panel of Figure 25 focusing on the three key variables (CD3, CD4, CD8β) with blue points from the control sample and the red points from the GvHD positive sample. We observe that, in addition to the high CD3, CD4, CD8β region, the distribution of the positive sample is different from the control

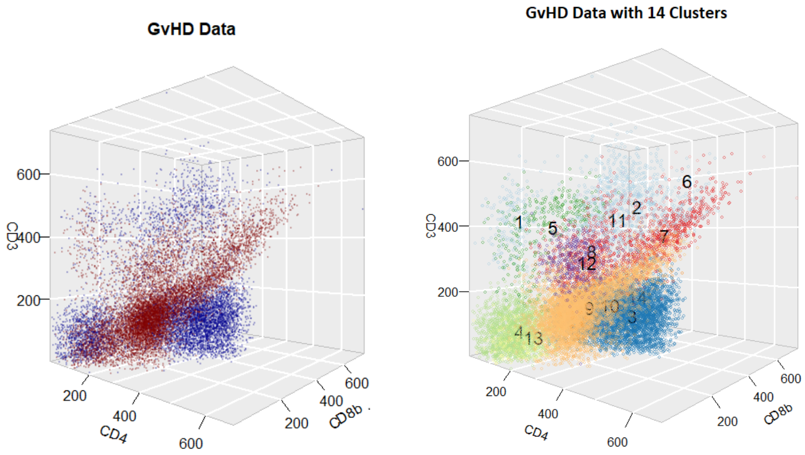


Figure 25: **Left:** 3D scatterplot of the positive sample (red) and the control sample (blue). **Right:** Final clustering result of combined GvHD data.

Table 1: Table of the cluster sizes and the weighted proportions of positive observations within each cluster.

Cluster	1	2	3	4	5	6	7
Size	202	948	3881	1859	338	17	812
Prop	.458	.343	.008	.296	.341	.000	.934
p-value	.30	7×10^{-20}	0	$3{\times}10^{-63}$	$4{\times}10^{-8}$	1×10^{-4}	$6{\times}10^{-103}$
Cluster	8	9	10	11	12	13	14
Size	468	6191	251	37	478	402	8
Prop	.690	.888	.673	.669	.794	.841	.310
p-value	$2{\times}10^{-13}$	0	$1{\times}10^{-6}$	.09	$6{\times}10^{-30}$	3×10^{-33}	.52

A proportion 0.5 indicates that the two sample has equal proportion in the region. The p-value is the simple proportional test to examine if the two sample has equal proportion in that cluster.

sample also in some region with medium to the low CD3, CD4, and CD8β. Later we will demonstrate that our clustering framework can identify all such differences in distributions.

To apply the skeleton clustering for a fair comparisons for the two samples, we first construct knots from each sample separately. Specifically, we apply the k-means method to find $k_1 = [\sqrt{n_1}]$ knots for the positive sample and find $k_2 = [\sqrt{n_2}]$ knots for the control sample. This ensure that both sample are well-represented by knots. We then combine the two samples into one dataset and combine the two sets of knots into one set with $k_1 + k_2$ knots. We create edges among the combined knots and apply the Voronoi density (VD) to measure the edge weights. To segment the knots, we use average linkage criterion because the clusters can be overlapping and our empirical experience suggests average linkage for this scenario. The skeleton clustering result is displayed in the right panel of Figure 25 with the number of final cluster chosen to be $S = 14$ [Baudry et al., 2010].

For further insights, we examined the weighted proportion of positive observations in each cluster. A proportionally smaller weight is assigned to each positive observation to accommodate the fact that there are more positive observations ($n_1 = 9083 > n_2 = 6809$). After such normalization a weighted proportion of 0.5 means that the positive and control observations are balanced in one region. A summary of the weighted proportion of clusters is presented in Table 1. We note that clusters 7,9,12, and 13 are majorly composed of positive observations (proportion > 0.75), and clusters 3 and 6 are majorly composed of observations from the control sample (proportion < 0.25). We also include the p-value for testing if the the proportions equal 0.5. Admittedly, because we use the data to find clusters and use the same data to do the test, the p-values in Table 1 may tend to be small.

Clusters with majorly positive observations and clusters with majorly control observations are depicted in the two panels in Figure 26. Cluster 7 corresponds to the high CD3, CD4, CD8β region identified by previous works with nearly all data points belonging to the positive patient. Cluster 6 is also scattered in the high CD3, CD4, CD8β region but has all the observations coming from the control sample. However, the small size (only 17 data points) of Cluster 6 makes unclear if it is a real structure or due to pure randomness. Overall our method succeed in identifying the CD3+ CD4+ CD8β+ area for the GvHD positive patient like the previous model-based clustering approaches. Note that the data we are using are two individuals from the original 31 individuals in the GvHD study, which does not account for the inter-individual variability.

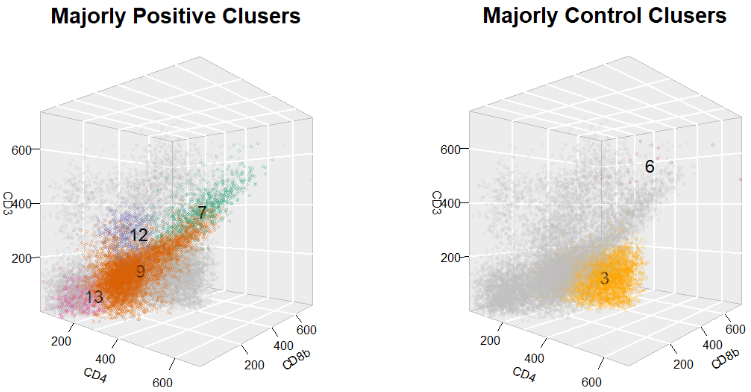


Figure 26: Clusters with majorly positive observations and majorly control observations

Our clustering approach have some additional findings. Cluster 9, 12, and 13 also have high proportion of positive samples. These clusters are in mid to low CD3, CD4, CD8β region. For the control case, in addition to the small Cluster 6, Cluster 3 is a large cluster with nearly all the observations are from the control sample. It is located in the high CD8β but low CD3 and CD4 region.

Model-based clustering approaches Lo et al. [2008], Baudry et al. [2010] have an advantage for managing this cytometry data as they can parametrically describe the behaviors of data samples in different regions. The overlapping between different structures and the overall 4-dimensional feature space are also applicable with model-based clustering methods. However, the proposed skeleton clustering approach can result in graphical representation of each clusters that can be visualized for intuitive understanding. We include the skeleton graphs of the GvHD data clusters from the proposed clustering approach in Appendix D.9. Moreover, model-based approaches can still be limited to some regular shapes of the clusters in the ambient space, while applying the proposed clustering method helps identify clusters with complex structures. Cluster 9, for instance, shows a hammer-like structure based on the skeleton representation (see Figure 16).

Our results suggest a potential procedure for diagnosing GvHD. Biomarkers from a new patient can be divided into clusters with respect to the learned segmentation, and doctors can mainly focus on the sample points that fall into regions 3, 7, 9, 12, and 13. If the patient has many points in Clusters 7, 9, 12, and 13, the patient likely has GvHD. Note that our current result is only based on two individuals and, with a descriptive purpose, is not accounting for the variability between different individuals and different cases. To use it for practical diagnosis, a more comprehensive analysis based on a larger and more representative sample is required.

F.2 Zipcode Data

This dataset consists of $n = 2000$ 16×16 images of handwritten Hindu-Arabic numerals from [Stuetzle and Nugent, 2010]. We use the overfitting k-means to find $k = 45$ knots. Similar to the procedure in Section 5, we consider four similarity measures to obtain the edge weight: VD, FD, TD, and AD. We use single linkage for the the four skeleton clustering approaches and compare them to three traditional methods: the direct single linkage hierarchical clustering (SL), the direct k-means clustering (KM), and spectral clustering (SC).

The result is shown in the left panel of Figure 27 with the adjusted Rand index plotted against different number of total cluster S. The gray vertical line indicates $S = 10$, which is the actual number of digits. The skeleton clustering with VD (Voron) gives the best clustering result in terms of adjusted Rand index at the true 10 clusters and gives good clustering results when the number of clusters is specified to be larger than the truth. However we note that spectral clustering (SC) and naive k-means clustering (KM) give comparably good results with small number of clusters.

The right panel of Figure 27 is the "denoised" version of the digits. We estimate the density of each observation by $[\sqrt{n}]$-nearest-neighbor density estimator and remove the observations with the lowest 10% density. We see that all clustering results are slightly improved, but such improvement may come from the decreased total sample size after denoising. Notably, the skeleton clustering with Tube density (Tube) generates significantly better clustering results after denoising the data, giving adjusted Rand indexes comparable to skeleton clustering with Voronoi density. This shows skeleton clustering with Tube density can be sensitive to noises in real data but still has the potential to give insightful clustering results.

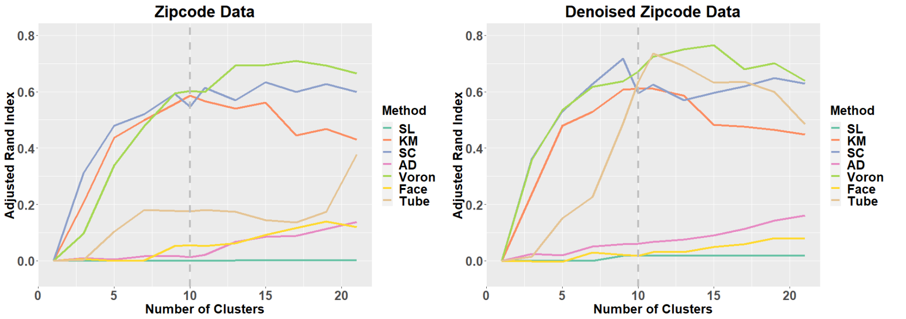


Figure 27: Comparison of different similarity measures on all Zipcode Data.

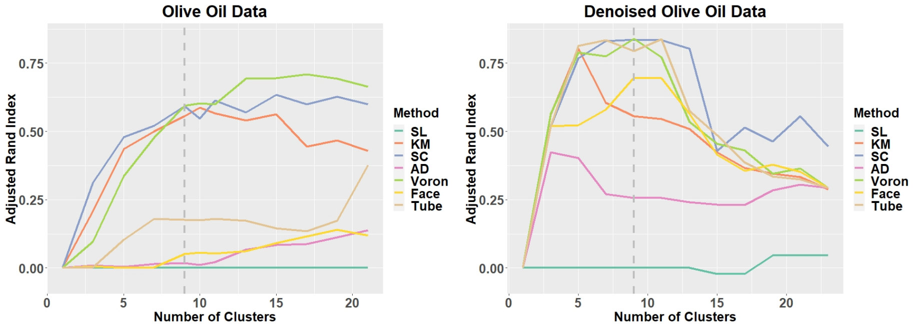


Figure 28: The clustering performance under different number of final clusters of the Olive oil data.

F.3 Olive Oil Data

We consider another real dataset: the the Olive Oil data [Tsimidou et al., 1987], a popular dataset for cluster analysis. This data set represents $d = 8$ chemical measurements on different specimens of olive oil produced in 9 different regions in Italy (northern Apulia, southern Apulia, Calabria, Sicily, inland Sardinia, and coast Sardinia, eastern and western Liguria, Umbria) . There are a total of $n = 572$ observations in the dataset.

Same comparison procedure as in Section F.2 is employed. The performance of different similarity measures is presented in Figure 28. Different color denotes different similarity measures and the gray vertical line indicates the actual number of clusters 9. Overall, the skeleton clustering with Voronoi density and Tube density works well; the spectral clustering also performs well in this case. The fact that average distance fails to capture clusters in the data highlights the importance of using a density-aided similarity in this case. Note that we also include the clustering performance on the 'denoised' data, in which we remove the 10% observation with the lowest $\sqrt{n}$-Nearest-Neighbor density estimate.